

福井大学審査
学位論文[博士(工学)]

多視点撮影像に基づく任意視点映像
生成に関する研究

2009 年 3 月

梁 華剛

目 次

第一章 序 論	1
1.1 本研究の背景	1
1.2 本研究の目的	4
1.3 本論文の構成	5
第二章 射影幾何学の理論	6
2.1 ユークリッド幾何と射影幾何	6
2.2 ピンホールカメラモデル	7
2.3 エピポーラ幾何	8
2.4 Fundamental Matrix (F 行列)	10
2.5 Homography	12
第三章 View Morphing	15
3.1 モーフィング	15
3.2 View Morphing の基本的な考え方	16
3.3 View Morphing の問題点	19
第四章 2 視点画像に基づく View Morphing に対する視点自由度の 拡張	20
4.1 基本的な考え	20
4.2 視点自由度の拡張	20
4.3 視点自由度特殊な視点設置の場合の拡張	24
4.4 本手法の適用条件	31
4.5 シミュレーション実験と考察	31
4.5.1 各軸方向の平行移動の影響	32
4.5.2 各軸回りの回転の影響	32
4.5.3 複数のずれの組み合わせによる影響	36

第五章 任意視点画像の生成	40
5.1 基本的処理の流れ	40
5.2 具体的処理	40
5.2.1 対応点の指定	40
5.2.2 平行化視点画像の生成	42
5.2.3 三角形パッチの生成	43
5.2.4 平行化任意視点画像上の対応点座標の算出	45
5.2.5 テクスチャマッピング	45
5.2.6 最終的任意視点画像への変換	46
5.3 実験	47
5.4 考察	51
第六章 まとめ	52
参考文献	53
謝辞	54

第一章 序 論

1.1 研究の背景

近年、映画やテレビなどの娯楽産業では実画像に基づく画像合成に対する需要が高まってきている。さらには、近年のブロード・バンドの発達やデジタル放送化におけるコンテンツ不足などから、スポーツ中継やコンサートなどをユーザが任意の視点から観戦するビュー・オン・デマンドなどへの期待もある。例えば、米国CBSテレビが2001年のスーパー・ボウル中継で実際に放送に用いた“Eye-Vision”システムでは、スポーツ中継において視点位置を任意に動かすことの有効性を示したと言える。また、1999年のワーナー・ブラザーズの映画“The Matrix”で用いられたシステムでは、時間が静止した状態で視点が動くようなコンピュータ・グラフィクスが実画像を元にして生成され、大きな話題を呼んだ。このように、実画像からの画像合成システムとして実用化されているものが増えてきているが、それらのシステムではグラフィクス生成に多大なコストが必要とされている。たとえば先述の映画“The Matrix”におけるコンピュータ・グラフィクスでは、たかだか60秒のシーンのために膨大な費用が投入され、大量の人手による数ヵ月におよぶ作業の結果として非常に良好なグラフィクスが生成されている。

一方、コンピュータ・ビジョンの分野において、画像からシーン内の物体の3次元形状を再構成する問題は一般的であり、その中でも複数カメラからの入力画像を用いた手法はステレオ・ビジョンと呼ばれている。最も基本的なステレオ・ビジョンの原理は、あらかじめ各カメラのキャリブレーションを行い、撮像画像中の対応点を探索することにより、それらの対応点の逆投影線の交点の3次元座標を計算するというものである。ステレオ・ビジョンによって正確な3次元形状を再構成できれば、いくつかのカメラでシーンを撮影することにより、推定された各物体や各人物の3次元形状をポリゴン・モデル等のコンピュータ・グラフィクスで表示し、冒頭で述べたような需要に対処することが可能である。しかしながら、実際のステレオ・ビジョンにはキャリブレーションを行うことは煩雑であり、また高精度のキャリブレーションを行うことは難しい問題である。

近年、コンピュータ・グラフィクスやコンピュータ・ビジョンの分野においてイメージ・ベースド・レンダリング(Image-Based-Rendering: IBR) と呼ばれる画

像合成方法を用いる場合が増えている。従来のコンピュータ・グラフィックスの多くは物体の3次元形状や物体表面の反射係数、光源情報などのモデルを明示的に与えることにより実現されているので、それらモデル・ベースの画像合成法と対比して、入力画像から画像を合成するための手法を総称してIBR と呼ぶものである

[1, 2, 3, 4, 5, 6, 7, 8]。IBR の最も単純な例としては、モーフィング手法が挙げられる。モーフィング手法とは、複数の画像の間で検出された対応点について、画像平面上の2次元で対応点の座標や画素値を内挿する画像合成法である[1, 2, 4]。モーフィング手法で合成された画像はカメラ光学や物体の3次元形状などを考慮せずに画像平面内だけで座標位置が線形内挿されてしまうため、一般的には射影幾何学的な正確さが失われてしまう。しかし、カメラ配置を適切に行い、画像間の対応点の精度も高ければ、多くの場合は人間の目には不自然さを感じさせない画像合成が可能である。そのため、2 画像間で歪みのない画像合成を行う“View Morphing” と呼ばれる手法がSeitz らによって提案されている[1]。しかしながら、Seitz 手法は、カメラ間の直線上から外れた視点の合成は原理的に不可能であった。そこで、本研究では、未校正のカメラ群から得られた多視点撮影画像から、3次元モデルは復元せずに視点位置設置の自由度の高い任意視点画像を生成することを考える。

一方、モデルベースの方法として、ステレオ・ビジョン方法がある。ステレオ・ビジョンとは、三角測量と同様の原理による3次元計測手法である。コンピュータ・ビジョンの分野において、カメラ・モデルとしてピンホール・カメラが一般的に用いられている。ピンホール・カメラとは焦点および画像平面によってモデル化されるカメラであり、図 1.1 に示すようなモデルである。ピンホール・カメラでは、撮影の対象シーンはカメラ焦点に向かって投影され、その投影線と画像平面の交点へと投影される。このようにして投影された点の集合が画像である。逆に、画像中のある画素で撮影された点は、焦点と画素を結んだ投影線の延長上に存在すると言える。

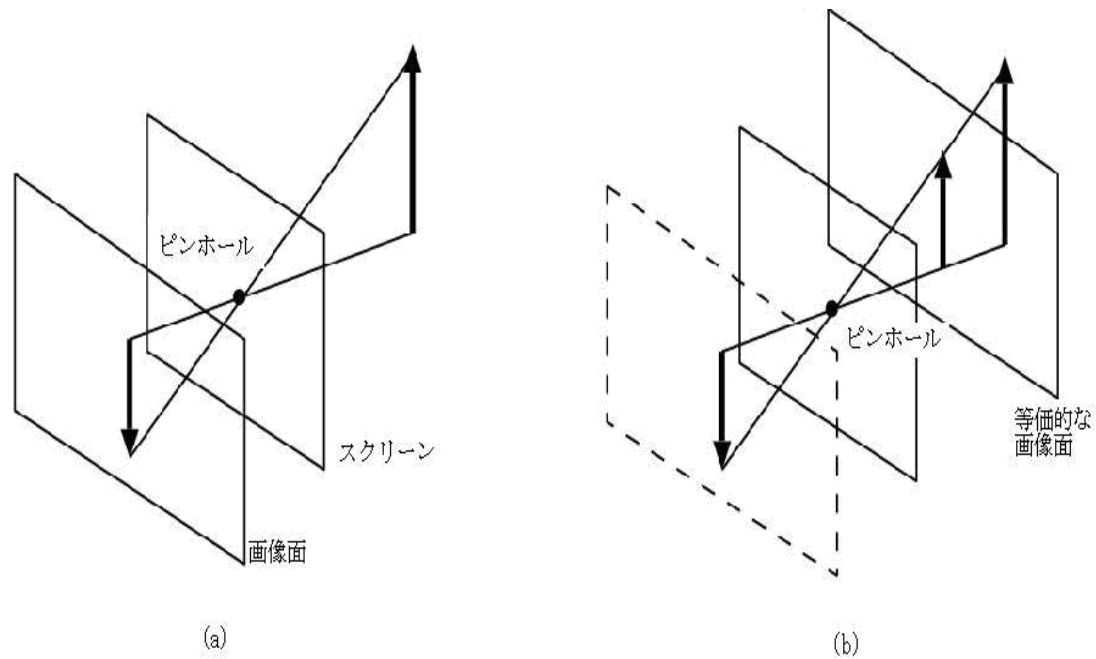


図 1.1：ピンホールカメラと中心投影モデル

2つのカメラで撮影を行った場合、シーン内のある1点は、各画像平面に投影されることになる。そこで、あらかじめ2つのカメラの画像平面や焦点の配置などが正確にわかっているならば、画像処理によって画像中の対応点を求め、それらの点を撮影対象シーンの3次元空間へ逆投影することによって、それらの逆投影線の交点に物体表面が存在することが推定できる。このようにして、画像中の各画素について画像間の対応点を求めることにより、シーン内の3次元情報を再構成するのがステレオ・ビジョンの原理であるが、ステレオ・ビジョンにはいくつかの難点が存在する。

カメラの画像平面や焦点に関する情報を求める処理はカメラ・キャリブレーションと呼ばれる。カメラ・キャリブレーションを行うことは煩雑であり、また高精度のキャリブレーションを行うことは難しい問題である。ただし、キャリブレーションは以下のような作業によって行われるものである。まず、あらかじめ形状既知な物体などをシーン内に配置し、世界座標系で記述されるシーン内の3次元座標位置と、その点が画像平面に投影された場所を記述した画像座標系での2次元座標位置の対応関係を求める。このような対応関係を与える点をキャリブレーション点と呼ぶ。キャリブレーション点を大量に与えることによって、大量の3次元—2次元の

対応関係を獲得し、それらの対応関係からカメラの各パラメータの推定が行われる。最も理想的なピンホール・カメラの場合、たかだか11個のパラメータでカメラ・モデルを記述することができる。したがって、理論上は6点のキャリブレーション点からカメラ・モデルの同定が可能であると言える。しかしながら、実際に画像平面で求まる2次元座標はサンプリングされた後のデジタル画像における座標であるため、キャリブレーション点の画像中の座標位置には誤差が含まれる。また、実際問題としてキャリブレーション点の3次元座標を正確に求める（あるいは正確な3次元座標位置にキャリブレーション点を配置する）ことも困難である。さらに、最終的なカメラ・パラメータ推定処理において、キャリブレーション点が撮影対象シーンとなる3次元空間中でなるべく広範囲にわたって存在しないと、推定誤差が大きくなることも知られている。その他、理想的なピンホール・カメラと実際のカメラの違いに起因する誤差要因などが数多く存在するため、ステレオ・ビジョンに耐え得る精度のキャリブレーションを行うためには、およそ数十～数百の対応関係が必要とされる。そこで、本研究では、キャリブレーションを行わず、高品質な任意視点画像を生成する方法を考える。

1.2 本研究の目的

前節で述べたように、ステレオ・ビジョンを精度良く実行することは、困難な問題である。そこで、より精度を高める目的やより完全な3次元形状を求めるための目的で、より多くのカメラを用いるような研究が提案されている。多数のカメラを用いることにはいくつかの利点がある。たとえば、カメラの数を増やすことにより、どのカメラからも撮影されないような死角を減らすことが可能である。しかしながら、ステレオ・ビジョンに用いるカメラはキャリブレーションを行う必要があるため、カメラの数が増すにつれ、キャリブレーションの煩雑さも増すことは明らかである。

一方、Seitzらが提案するカメラ間の幾何学関係を正確に反映したView Morphing方法では、2つの視点を結ぶ直線上の線形補間によって任意視点の画像を生成するので、視点位置がその直線上に限られるという問題がある[2, 4]。

そこで、本研究では、未校正のカメラ群から得られた多視点撮影画像から、3次元モデルは復元せずにより自由度高い任意視点画像を生成することを考え

る。そこで、View Morphing での視点位置の制約を本文の方法によって緩和し、空間中の任意視点での画像を生成するものである。これにより、2 視点の撮影画像から、入力視点を結ぶ直線間だけではなく平面上でも任意視点からの像を合成することができる。さらに、3 次元空間中の任意視点画像も生成することができる。

1.3 本論文の構成

本論文は六章で構成される。まず、第一章では序論として本研究の背景や目的や近年、複数台のカメラから自由な視点での画像を生成する方法に紹介し、次に、第二章において、本論文の基盤となっている射影幾何の理論について述べる。具体的には、射影幾何について2つの理論を用いており、その1つはエピソード拘束を記述するため fundamental matrix と呼ばれる行列を利用する、もう1つは平面射影を記述するための homography と呼ばれる行列を行う。第三章ではモーフィングと View Morphing の概念と理論および、その手法の持つ問題点を整理する。第四章では、2 視点画像に基づく View Morphing に対する視点自由度の拡張として、射影幾何関係に基づく任意視点の設置を空間中への拡張および提案の手法の幾何関係の妥当性のシミュレーション実験について説明する。それから、第五章では、本提案の有効性を確認するため、実撮影環境での任意視点画像を生成する実験と合成できた画像の有効性の検討について述べる。第六章では、本研究の結果をまとめ、今後の課題について述べる。

第二章 射影幾何学の理論

本章では、ユークリッド幾何と射影幾何概念およびその空間におけるカメラのキャリブレーションについて説明する。そして、本文中で用いている2種類の射影幾何理論について述べる。第三章のView Morphing手法と第四章の本文の拡張手法とも、この理論に基づき視点画像を平行化し、さらに同じ理論を用いてテクスチャマッピングを行う。

2.1 ユークリッド幾何と射影幾何

我々が一般的に慣れ親しんでいる座標系は、ユークリッド幾何で定義された X 、 Y 、 Z の直交3軸で記述される3次元座標である[2]。カメラ・パラメータを記述する際の最も単純な方法として、ある特定のユークリッド空間との関係で記述する方針が考えられる。古典的なステレオ・ビジョンでは、このような方針に基づいて、各カメラとワールド座標系の関係をユークリッド幾何で記述するようなキャリブレーションを行うことが一般的である。言うまでもなく、ユークリッド空間における3次元再構築のような場合には、ユークリッド空間におけるキャリブレーションが必須である。また、全カメラで共通の座標系が必要な場合全般において、このような統一的な記述法は有利であるとも言える。しかし、このような方法を用いた場合には、カメラ間の相対関係を記述するためには、一度そのユークリッド空間を経由して記述することになる。

ユークリッド幾何とカメラの間の関係を求めるためには、シーン内の3次元座標と、その投影点である画像中の2次元座標の対応点が大量に必要である。すなわち、ユークリッド幾何におけるキャリブレーションは、正確な位置が既知であるような点がシーン内に大量に必要であり、実作業としては煩雑なものとならざるを得ない。

例えば単にカメラ間で対応点探索を行うような場合、前述のような中間データの経路を用いずに、直接カメラ間の関係を記述することにより対応点探索に必要な拘束条件を求める理論があり、それらは射影幾何と呼ばれる。本論文では、2つの

射影幾何理論を用いている。1 つは、エピポーラ幾何を記述するためのfundamental matrix (F 行列) であり、もう1 つは、homography である。どちらも3×3 行列であるが、その目的や用途は異なるものである。射影幾何のキャリブレーションでは画像間の対応点のみが必要とされるため、実作業としては簡易なもので済むという利点がある。そのような違いを表すために、ユークリッド幾何におけるキャリブレーションは“強キャリブレーション(strong calibration)”、射影幾何におけるキャリブレーションは“弱キャリブレーション(weak calibration)” と呼ばれている。

近年のコンピュータ・ビジョンにおいて射影幾何を用いた研究が増えているが、その理由の1 つとして弱キャリブレーションが簡易な作業で行えるという点が挙げられる。例えばセルフ・キャリブレーションと呼ばれる、撮影とキャリブレーションを同時に行うようなシステムでは、画像間の対応点のみでキャリブレーションができることは大きな利点と言える。弱キャリブレーションの理論について次の節で説明する。

2.2 ピンホールカメラモデル

画像の投影変換を行うためには3次元空間における撮影画像の位置を正確に記述する必要がある。そこでカメラをモデル化することで、カメラと撮影画像の位置関係を捉える。ここでは、最も簡単なカメラモデルであるピンホールカメラモデルを使用して表すことにする。

ピンホールカメラとは、暗箱にあげた非常に小さな穴（この穴をピンホールと呼ぶ）を通して外界の光が入射し、穴の反対側の面の上に像が映るというものである。このとき、結像する全ての光はピンホールを通る。この原理に基づくカメラモデルをピンホールカメラモデルという。

ピンホールカメラモデルによる投影変換について幾何学的に述べるため、図2.1のような座標系を考える。このとき、実3次元空間中の点 $P(X,Y,Z)$ と画像の上の座標における投影点 $p(u,v)$ の間には、以下の関係が成り立つ

$$u = f \frac{X}{Z} \quad v = f \frac{Y}{Z} \quad (2.1)$$

ただし、画像上の x,y 軸はカメラ座標の X,Y 軸と平行で、 Z 軸が画像の中心

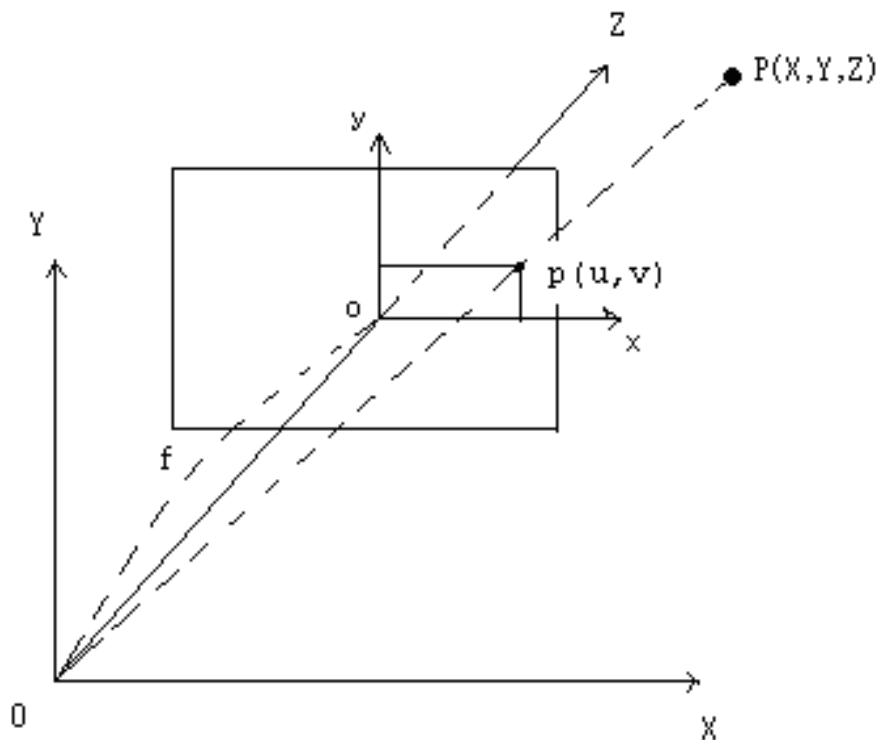


図 2.1 : 透視投影変換

を通り、原点から画像面までの距離（焦点距離）が f であるとする。

このとき、同じ X, Y の値を持つ点を考えると、視点（＝原点）からの奥行き（＝ Z 座標値）が大きいほど、画面の中央に投影されることがわかる。このため、視点から遠くにあるものほど小さく見えるという現象がシミュレートされていることが確認できる。このような投影変換を透視投影もしくは中心投影などと呼ぶ。本文の議論では、使用するカメラの投影変換にはこのピンホールカメラモデルによる透視投影変換を用いる。

2.3 エピポーラ幾何

2.2

カメラ・モデルとしてはピンホール・カメラを用いることが一般的である。ここで、ある対象シーンを2つのカメラで撮影した場合を想定すると、図2.2 に示したようなモデルが考えられる。その際、2カメラの焦点を結ぶ直線を想定すると、その直線を通るような任意の平面を定義することができる。このような平面はエピ

ポーラ平面と呼ばれる。エピポーラ平面は 360° の自由度を持つが、エピポーラ平面と画像平面が平行にならない限りは、任意のエピポーラ平面とそれぞれの画像平面は交線を持つ。そのような交線はエピポーラ線と呼ばれる。ある1つのエピポーラ平面は、それぞれの画像中に対応するエピポーラ線を決定する。また、図2.3に示すように、別カメラの焦点が画像平面上に投影された点はエピポールと呼ばれる。理論上、全てのエピポーラ線がエピポールを通ることはエピポーラ線の定義より明らかである。このようなカメラ間の相対関係の概念はエピポーラ幾何と呼ばれている。

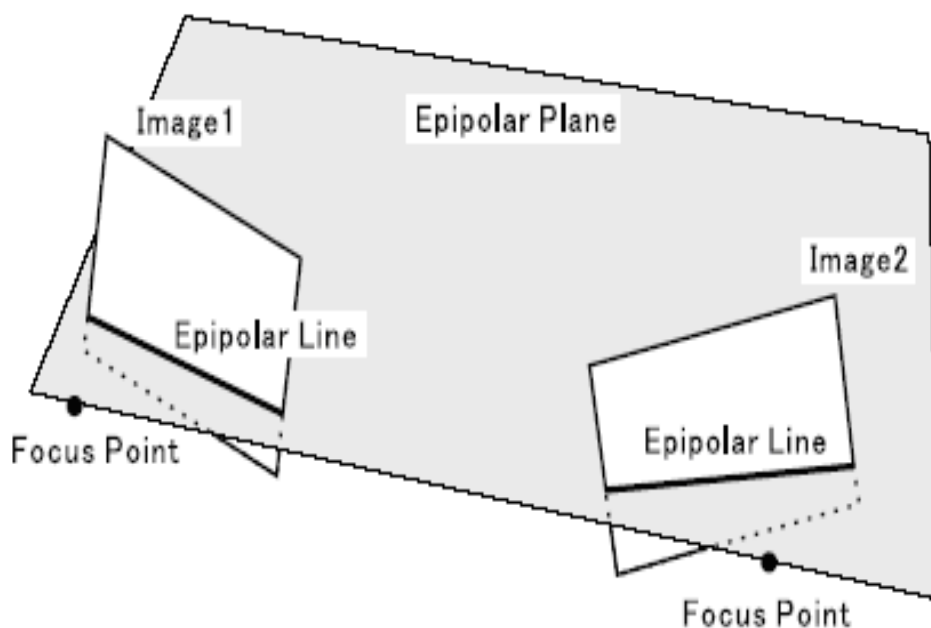


図2.2：エピポーラ幾何

2つの画像中に、対応するエピポーラ線が求まっているような状況を想定すると、エピポーラ線の定義より、それらのエピポーラ線は、3次元空間中の同一の平面が投影されたものである。従って、画像中の対応点探索を行う場合には、対応するエピポーラ線内だけで探索を行えば十分である。

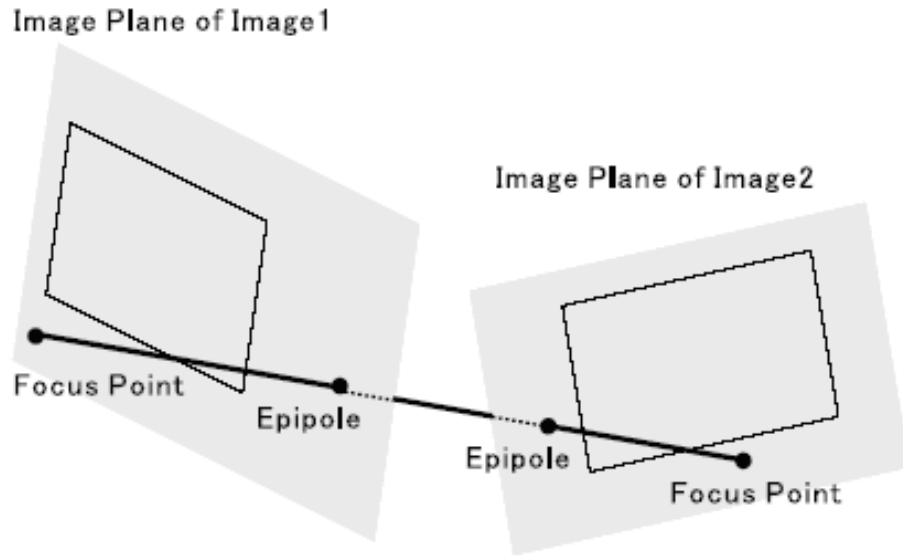


図2.3：エピポール

2.4 Fundamental Matrix (F行列)

シーン内の任意の点がそれぞれ撮影された画像座標の同次座標を

$m_0 = (u_0, v_0, 1)^T$ 、 $m_1 = (u_1, v_1, 1)^T$ とすると、理論上、

$$m_0^T F m_1 = 0 \quad (2.2)$$

の関係が成立する。この行列 F はfundamental matrix (F 行列) と呼ばれ、2つカメラ間のエピポーラ幾何を記述する。

ある2カメラ間で F 行列が求まると、図2.4 に示すように、両画像中の対応するエピポーラ線を任意に求めることができる。カメラ C_0 、 C_1 について、 C_0 から C_1 中にエピポーラ線を求める F 行列を F_a 、 C_1 から C_0 中にエピポーラ線を求める F 行列を F_b とすると、 C_0 の画像中の任意の点を同次座標で記述した m_0 と F 行列 F_a から、

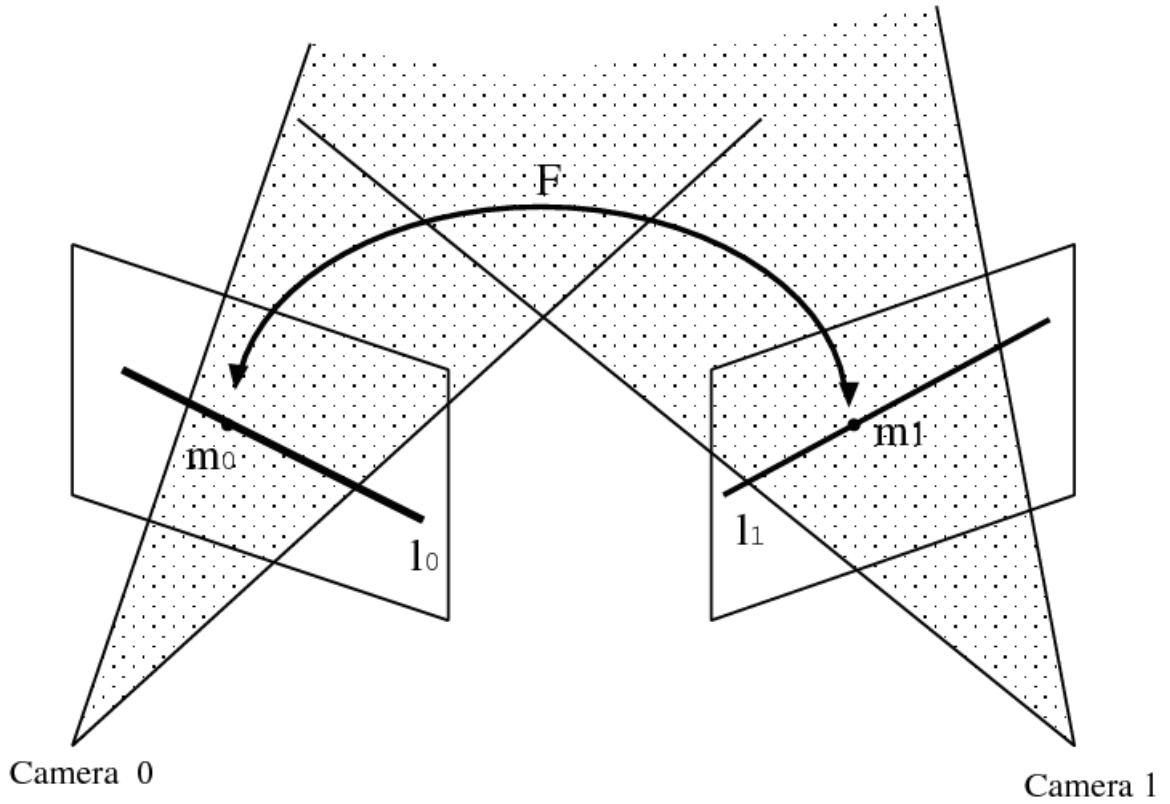


図2.4 : F行列による対応エピポーラ線の取得

$$l_1 = F_a m_0 \quad (2.3)$$

のようにして、点 m_0 に対応するエピポーラ線 $l_1 = (l_{11}, l_{12}, l_{13})^T$ を C_1 の画像中に求めることができる。当然、直線 $l_0 = (l_{01}, l_{02}, l_{03})^T$ 上の任意の点 m_1 と F_b から、

$$l_0 = F_b m_1 \quad (2.4)$$

として C_0 の画像中のエピポーラ線を求めることが可能であるため、対応する任意のエピポーラ線 l_0 と l_1 をそれぞれの画像中に求めることが可能である。

なお、理論上 F_a と F_b には

$$F_a = F_b^T = F \quad (2.5)$$

の関係が成立するため、 F_a と F_b はカメラ C_0 、 C_1 それぞれの画像 a 、 b 間の対応点から同時に求まる。

また、画像 a 中のカメラ b のエピポール e_b と、画像 b 中のカメラ a のエピポール e_a について、

$$F_a e_b = 0 \quad (2.6)$$

$$F_b e_a = 0 \quad (2.7)$$

が成立するので、 e_b は F_a の最小の固有値に対応する固有ベクトルとして求まる。従って、 F 行列が求まると、エピポールの位置も求まったこととなる。

F 行列は 3×3 なので、9要素を持つが、 F 行列には定数倍の不定性があり、さらに F 行列の行列式について $\det F = 0$ という条件が存在するため、理論上、 F 行列は7自由度しかない。 F 行列の同定は画像中の既知の対応点から行われる。非線形解法を用いた F 行列の推定には7組の対応点が必要であり、線形解法では8組の対応点が必要であるが、実際には7、8点程度の入力では誤差が大きいことが知られている。しかしながら、例えばLED などのように検出が容易な点を対象シーン内で動かすことにより、必要な対応点を精度良く求めることは可能であるので、いずれにせよ強キャリブレーションに比べて簡易な作業であることは明らかである。

与えられた対応点から F 行列を推定する具体的な方法については、数多くの手法が提案されている。本論文で F 行列を用いる目的は、各画像中に対応するエピポーラ線を求めることである。従って、対応点とエピポーラ線の距離が十分に小さければ、 F 行列の精度は十分である。

2.5 Homography

図2.5 のように、3次元空間中のある同一の平面を2つのカメラで撮影した場合

を想定し、その平面上の任意の点がそれぞれのカメラで撮影された画像座標の同次座標を $m_0 = (u_0, v_0, 1)^T$, $m_1 = (u_1, v_1, 1)^T$ とすると、 3×3 の行列 H とスカラー s を用いて、

$$sm_0 = Hm_1 \quad (2.8)$$

という関係が成立する。この行列 H はhomography と呼ばれる平面射影行列である。

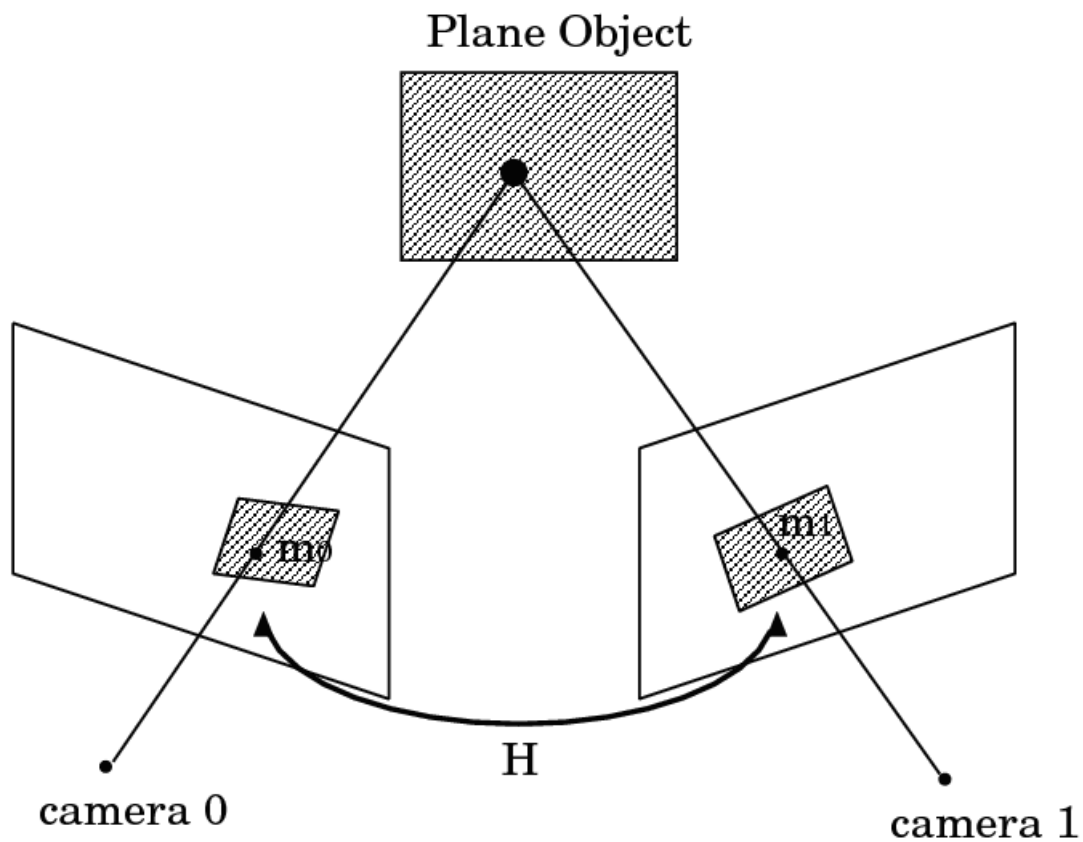


図2.5 : Homography

ここで、行列 H は、アフィン変換行列 H_α と扇形変形行列 B の積によって

$$H = H_\alpha B \quad (2.9)$$

と表せる。扇形変形行列 B とは図2.6 のような変形を表す行列であり、

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \alpha & \beta & 1 \end{bmatrix} \quad (2.10)$$

として表される、自由度2の行列である。また、アフィン変換行列 H_α は回転と並進を含む行列であり、6自由度を持つ。従って、行列 H の自由度は8である。具体的には、 H は 3×3 の行列であるが、スケールを任意に設定できるので、8自由度の行列として扱うことが可能となる。

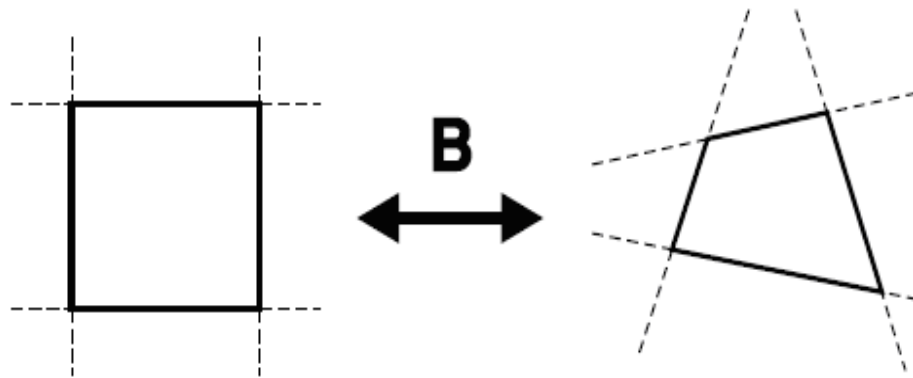


図 2.6 : 扇形変形行列

H_α による変換では、平行線は平行線に変換される。しかし、 B による変換では、直線は直線に変換されるものの、平行性や直角性は保存されない。したがって、 H による変換においても、平行性や直角性は保存されないものの、直線性は保存されることが理論上保証される。

平面上の4点の対応が両画像中で与えられれば、homography を求めることが可能である。ひとたびhomography が求まれば、その平面上の任意の点について両画像間で対応付けることができる。

以降の議論では、これら基礎行列 F と Homography を視点画像の平行化と、テクスチャマッピングに利用する。

第三章 View Morphing

本章では、まず、ある画像から別の画像へ滑らかに変化させるために、その中間を補う画像群を生成する技術であるモーフィングについて説明し、つぎに、2 視点の撮影画像からカメラ間の幾何学関係を正確に反映した補間画像生成法である View Morphing の概念とそれ実現するための幾何学変換について述べる。そして、Seitz らの提案した View Morphing 方法および、Xiao ら拡張の 3 視点 View Morphing 方法の問題点について説明する。

3.1 モーフィング

Morphing という言葉は “move” (動き) と “morphology” (形態) から生まれた造語であるという説と、“metamorphosis” (変形、変態) から生まれた造語であるという説がある。Morphing とは、ある形状から別の形状へ徐々に変化していく様子を動画で表現するために、その中間を補うための画像を作成する技術である。変化前後の画像の手動あるいは自動で定めた対応関係から、中間の画像を作り出すことで、滑らかに変化するモーフィングの動画が得られる。例えば、子供の顔から大人の顔へと徐々に変化させたり、原始人の姿から現代人の姿へと少しずつ変形していくアニメーションなどがモーフィングによって作られている。

例として、画像 A から画像 B へなめらかに変化させるモーフィングの処理具体的な流れを以下に示す。

1. 画像 A、B に複数の対応点を指定する。
2. 対応点群から三角形もしくは四角形パッチを構成する。
3. 中間画像での対応点群の位置を、合成比率を用いた線形補間により求める。
4. 得られた対応点群での三角形もしくは四角形パッチに対して、画像 A、B のテクスチャを合成比率と同じ比率で混合し貼り付け、最終的な中間画像を生成する。

この方法により、比率を 0 から 1 に徐々に変化させることで、画像 A から

画像 B までの連続的に変わる中間画像群を生成することができる。そして、それらの画像を連続表示することで、画像 A から画像 B へ滑らかに変化していく映像を生成できる。ほとんどのモーフィングでは、このように中間画像での対応点の座標とテクスチャの画素値を合成比率を用いた線形補間により求めている。

しかし、あるシーン状況を異なる 2 視点で撮影した画像をつかって、その中間の画像を生成したとしても、単に対応点と合成比率のみを用いるだけでは、一般的な場合、入力画像間の幾何学的関係を正確に反映したものにはならない。つまり、中間画像での物体の見えたかを崩してしまう。ある物体が違う物体に変形する映像を生成したいならば、物体の見え方は気にすることはないが、同じ物体で視点が変わっていく映像を生成したいならば、物体の見え方が保たれていないと不自然な映像になってしまう。そこで、この問題を解決する方法として View Morphing が提案されている。

3.2 View Morphing の基本的な考え

View Morphing とは、Seitz らが提案したカメラ間の幾何学関係を正確に反映したモーフィング技術である。基本的考え方は、入力画像を平行化させてモーフィングを行うことで、入力視点画像間の幾何学的関係を正確に反映できるものである。平行化画像を生成するために、透視投影変換による撮影画像を平行な 2 平面に投影して、平行化画像を作る。以下、その基になる透視投影変換のためのカメラモデルである。ピンホールカメラモデルと透視投影変換について述べ、View Morphing 処理の手順について述べる。

View Morphing 処理の手順は図 3.1 のように、以下の 3 つのステップでモーフィングを行う [6]。

1. 入力画像（図3.1）を光軸が平行でかつ焦点距離が同じになる仮想平面へ投影し、平行化入力画像（図3.2） I_0, I_1 を生成する。この処理はhomography によって実現される。
2. 平行化入力画像 I_0, I_1 の間でモーフィングを行い、平行化中間画像（図 3.3） I_s を生成する。
3. 平行化中間画像 I_s を逆投影し、中間画像（図 3.1）を生成する。

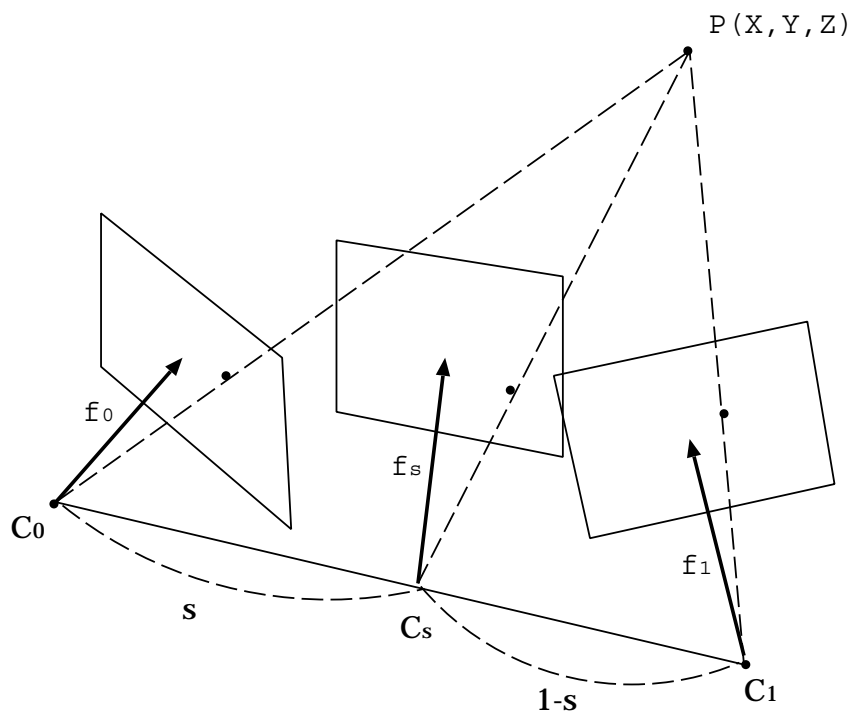


図 3.1 : 平行でない入力画像

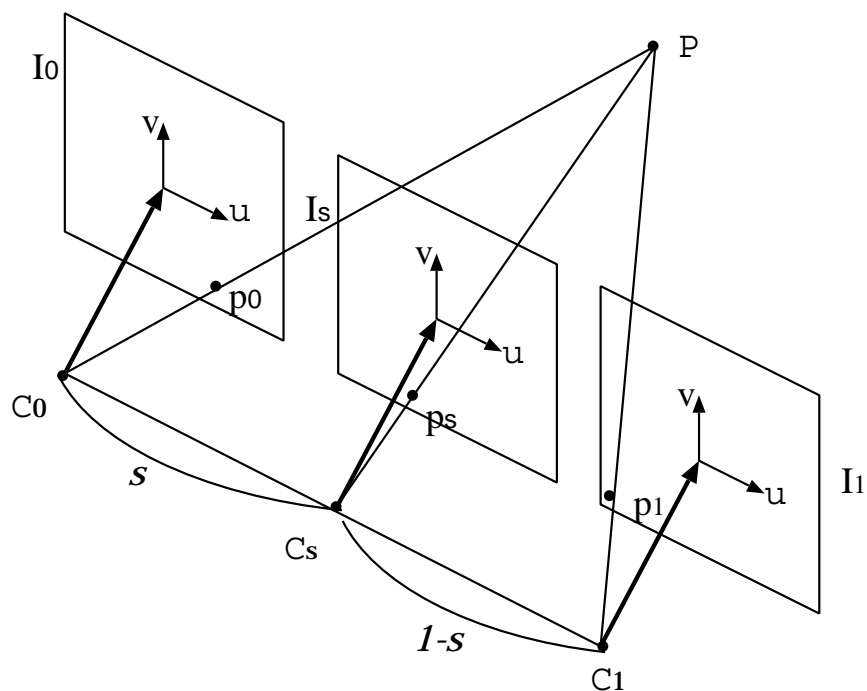


図 3.2 : 平行化入力画像

この3ステップの処理で、2つの入力画像を光軸が平行となる仮想平面上へ投影することで、平行化入力画像 I_0, I_1 間で幾何学的関係を正確に反映させることができる。

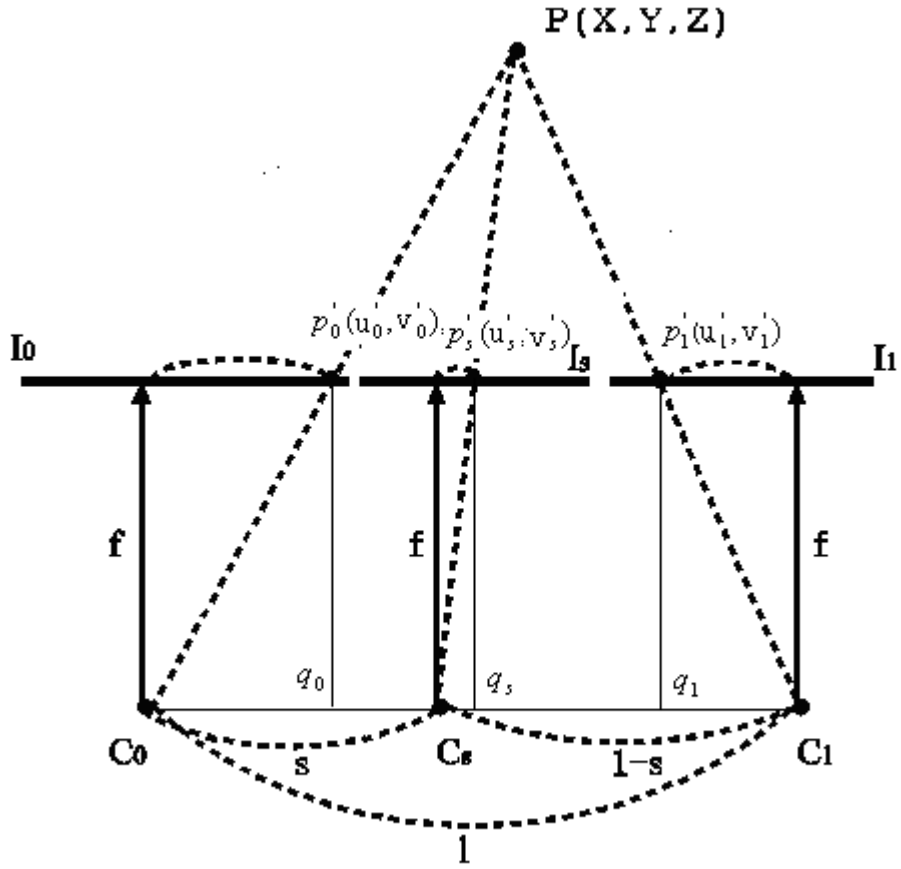


図 3.3 : X 軸と Z 軸だけの平行化入力画像

そこで、幾何学的関係を正確に反映していることを証明する。まず、図 3.3 の X 軸と Z 軸だけに注目すると図 3.3 のようになる。2 つの入力画像 I_0, I_1 の視点を C_0, C_1 とし比率 s で内分する任意視点を C_s とする。

図 3.3 に示すように、三次元空間中ある点 $P(X, Y, Z)$ 対応する各平行化画像上投影点は $p_0'(u_0', v_0'), p_s'(u_s', v_s'), p_1'(u_1', v_1')$ となる。光軸が平行でかつ焦点距離が同じなので $\overline{p_0'p_s'} : \overline{p_s'p_1'}$ は $s : (1-s)$ になる。そして、 p_0', p_s', p_1' から $\overline{C_0C_1}$ に垂線を引き、その交点を各々 q_0, q_s, q_1 とし、 $\overline{C_0C_1}$ の距離を l とする。さらに、 C_0 を原点とし C_1 の方向を正とすると、 $\overline{q_0q_s}, \overline{q_sq_1}$ は、

$$\overline{q_0q_s} = (sl + u_s') - u_0' \quad (3.1)$$

$$\overline{q_sq_1} = (1-s)l + u_1' - u_s' \quad (3.2)$$

で表せる。そして、 $\overline{p'_0 p'_s} : \overline{p'_s p'_1} = \overline{q_0 q_s} : \overline{q_s q_1}$ より、

$$s : (1 - s) = (sl + u'_s) - u'_0 : (1 - s)l + u'_1 - u'_s \quad (3.3)$$

となり、この式を解くと以下の式が導き出される。

$$x'_s = (1 - s)u'_0 + su'_1 \quad (3.4)$$

さらに、 y 軸に対しても同様に考えれば平行化中間画像 I'_s の対応点 p'_s の位置ベクトルを、合成比率 s を用いた線形補間により以下の式で求めることができる。

$$p'_s = (1 - s)p'_0 + sp'_1 \quad (3.5)$$

これにより、幾何学的に正確な中間画像を生成することが可能となる。また、視点 C_s は 2 つ視点 C_0, C_1 の線分を比率 s で補間したものであるため、正確な視点位置がわかっていなくてもその中間画像を比率の意味で正しく生成できることもわかる。

3.3 View Morphing の問題点

Seitz らの提案した View Morphing では、カメラの数が 2 つであるということに依存した処理を用いており、仮想な任意視点の位置は 2 視点カメラの中間の視点直線上に限定されている。また、その拡張による 3 視点 View Morphing では、任意視点の自由度は 3 視点カメラで囲まれた三角形の内部だけ可能であった [1, 2, 4]。

本研究では、射影幾何学関係を利用して、平行化視点画像間の対応点関係を導く。そこで、本研究では、高い自由度を持つ空間中の任意視点の画像を生成する手法を提案する。

第四章 2 視点画像に基づく View Morphing

に対する視点自由度の拡張

任意視点の位置は2視点のカメラの視点直線上に限定されないようにするために、本章では、2視点画像に基づく View Morphing に対する、視点自由度を拡張する手法を述べる。

4.1 基本的な考え方

一般的な配置の2台のカメラを用いた任意視点画像合成を行うモーフィング手法では、合成画像に幾何学的歪みが生じることが Seitz らによって指摘されている。この問題に対して Seitz らはカメラ間の幾何学関係を正確に反映する View Morphing 手法を提案した。

しかし、この手法では、仮想な任意視点の位置は2台カメラの視点結ぶ直線上に限定されている。この問題解決するために木村らと Xiao らは、3視点画像を用いた View Morphing の拡張を提案している。しかし、仮想視点は3視点で困れた三角形内部のみに設定可能である。そこで、本論文では、さらに高い自由度を実現するために2視点の View Morphing において、カメラの設置方法を工夫することによる任意視点位置の自由度の拡張を考える。

4.2 視点自由度の拡張

ここでは、視点画像を平行化したから平行化の視点画像上の対応関係を拡張する。あるいは、同じ内部パラメータを持つカメラを、2つ画像平面を平行に配置する(図4.1)。視点 C_0 のカメラの座標系をワールド座標系とし、平行化視点 C_1 のカメラの光軸の向きを C_0 のカメラの光軸と平行におく、この時、視点 C_0 、 C_1 の座標は、それぞれ $C_0(0,0,0)$ 、 $C_1(l_x, l_y, l_z)$ とする。

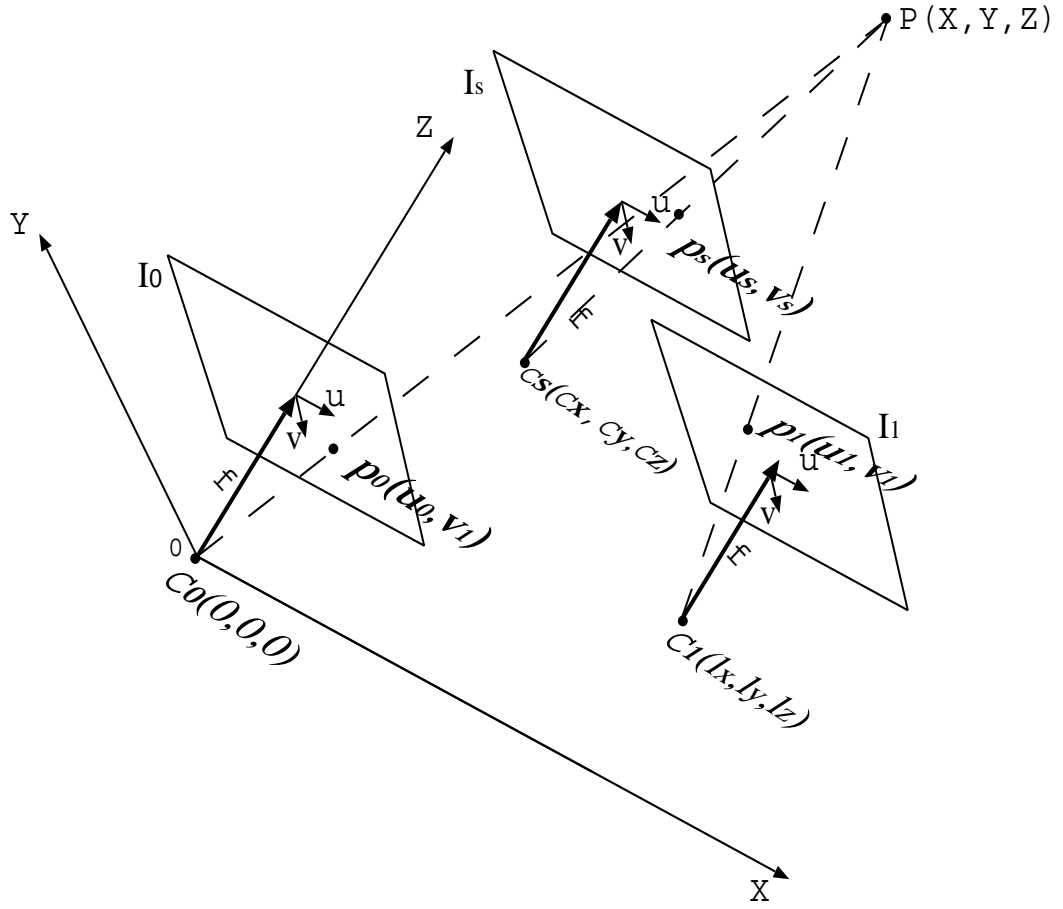


図 4.1 : 入力視点と任意視点およびその画像位置関係

この時、空間中のある任意の視点 C_s の座標を (C_x, C_y, C_z) とすると射影幾何関係により、ある点 P の空間のワールド座標 (X, Y, Z) と、各視点 C_i の画像 I_i 上の座標 (u_i, v_i) とは透視投影行列 P_i を用いて以下のように表せる。

$$w_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = P_i \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4.1)$$

ただし、 w_i は 0 でない定数であり、 $i = 0, 1, s$ とする。

ここで、カメラの光軸がそれぞれの画像平面の原点で直交し、焦点距離が共通の値 f であると仮定すると、入力視点 C_0 、 C_1 と任意視点 C_s のそれぞれに対する透視投影行列 P_0 、 P_1 、 P_s は、

$$P_0 = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (4.2)$$

$$P_1 = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -l_x \\ 0 & 1 & 0 & -l_y \\ 0 & 0 & 1 & -l_z \end{bmatrix} \quad (4.3)$$

$$P_s = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -C_x \\ 0 & 1 & 0 & -C_y \\ 0 & 0 & 1 & -C_z \end{bmatrix} \quad (4.4)$$

である。

この時、点 P に対応する各画像上での投影点の座標 (u_0, v_0) 、 (u_1, v_1) 、 (u_s, v_s) は、以下のように表せる。

$$\begin{cases} u_0 = \frac{X}{Z} f, \\ v_0 = \frac{Y}{Z} f \end{cases} \quad (4.5)$$

$$\begin{cases} u_1 = \frac{X - l_x}{Z - l_z} f, \\ v_1 = \frac{Y - l_y}{Z - l_z} f \end{cases} \quad (4.6)$$

$$\begin{cases} u_s = \frac{X - C_x}{Z - C_z} f, \\ v_s = \frac{Y - C_y}{Z - C_z} f \end{cases} \quad (4.7)$$

これらの式から、 X, Y, Z を消去すると、 (u_s, v_s) は $u_0, v_0, u_1, v_1, l_x, l_y, l_z, f$ を用いて、表すことができる。

まず、式(4.5)より

$$\begin{cases} X = \frac{u_0 Z}{f}, \\ Y = \frac{v_0 Z}{f} \end{cases} \quad (4.8)$$

であり、式(4.8)を式(4.6)に代入すると、

$$\begin{cases} u_1 Z - u_1 l_z = Xf - l_x f \\ v_1 Z - v_1 l_z = Yf - l_y f \end{cases} \Rightarrow \begin{cases} u_1 Z - u_1 l_z = u_0 Z - l_x f, \\ v_1 Z - v_1 l_z = v_0 Z - l_y f \end{cases} \quad (4.9)$$

を得る。式(4.9)を整理すると、 $u_1 - u_0 \neq 0$ 時、 $Z = \frac{l_z u_1 - l_x f}{u_1 - u_0}$ であるので、

$$\begin{cases} u_s = \frac{u_0 Z - C_x f}{Z - C_z} = \frac{u_0 \frac{l_z u_1 - l_x f}{u_1 - u_0} - C_x f}{\frac{l_z u_1 - l_x f}{u_1 - u_0} - C_z} \\ = \frac{u_0 u_1 l_z - u_0 l_x f - C_x f(u_1 - u_0)}{u_1 l_z - l_x f - C_z(u_1 - u_0)}, \\ v_s = \frac{v_0 Z - C_y f}{Z - C_z} = \frac{v_0 \frac{l_z u_1 - l_x f}{u_1 - u_0} - C_y f}{\frac{l_z u_1 - l_x f}{u_1 - u_0} - C_z} \\ = \frac{v_0 u_1 l_z - v_0 l_x f - C_y f(u_1 - u_0)}{u_1 l_z - l_x f - C_z(u_1 - u_0)} \end{cases} \quad (4.10)$$

一方、 $v_1 - v_0 \neq 0$ 時、 $Z = \frac{l_z v_1 - l_y f}{v_1 - v_0}$ により、

$$\begin{cases} u_s = \frac{u_0 Z - C_x f}{Z - C_z} = \frac{u_0 \frac{l_z v_1 - l_y f}{v_1 - v_0} - C_x f}{\frac{l_z v_1 - l_y f}{v_1 - v_0} - C_z} \\ = \frac{u_0 v_1 l_z - u_0 l_y f - C_x f(v_1 - v_0)}{v_1 l_z - l_y f - C_z(v_1 - v_0)}, \\ v_s = \frac{v_0 Z - C_y f}{Z - C_z} = \frac{v_0 \frac{l_z v_1 - l_y f}{v_1 - v_0} - C_y f}{\frac{l_z v_1 - l_y f}{v_1 - v_0} - C_z} \\ = \frac{v_0 v_1 l_z - v_0 l_y f - C_y f(v_1 - v_0)}{v_1 l_z - l_y f - C_z(v_1 - v_0)} \end{cases} \quad (4.11)$$

ここで、 $l_z \neq 0$ の時、式(4.10)と式(4.11)は、以下の形に変形できる。

$$u_1 - u_0 \neq 0 \text{ 時、} \begin{cases} u_s = \frac{u_0 u_1 - u_0 \frac{l_x f}{l_z} - \frac{C_x f}{l_z} (u_1 - u_0)}{u_1 - \frac{l_x f}{l_z} - \frac{C_z}{l_z} (u_1 - u_0)}, \\ v_s = \frac{v_0 u_1 - v_0 \frac{l_x f}{l_z} - \frac{C_y f}{l_z} (u_1 - u_0)}{u_1 - \frac{l_x f}{l_z} - \frac{C_z}{l_z} (u_1 - u_0)} \end{cases} \quad (4.12)$$

$$v_1 - v_0 \neq 0 \text{ 時、} \begin{cases} u_s = \frac{u_0 v_1 - u_0 \frac{l_y f}{l_z} - \frac{C_x f}{l_z} (v_1 - v_0)}{v_1 - \frac{l_y f}{l_z} - \frac{C_z}{l_z} (v_1 - v_0)}, \\ v_s = \frac{v_0 v_1 - v_0 \frac{l_y f}{l_z} - \frac{C_y f}{l_z} (v_1 - v_0)}{v_1 - \frac{l_y f}{l_z} - \frac{C_z}{l_z} (v_1 - v_0)} \end{cases} \quad (4.13)$$

ここで、 $\alpha = \frac{C_x f}{l_z}, \beta = \frac{C_y f}{l_z}, \gamma = \frac{C_z}{l_z}, a = \frac{l_x f}{l_z}, b = \frac{l_y f}{l_z}$ とすると、視点位置が決ま

ると、 l_x, l_y, l_z, f は定数であるから、 a, b は定数になる。このことから、式(4.12)と式(4.13)は、任意視点画像上の対応点座標が3つ変数 α, β, γ と2つ定数 a, b より求められることを表している。しかし、定数 a, b が分からない場合、任意視点画像上の対応点座標が算出できない。

4.3 特殊な視点設置の場合の拡張

前節で、入力視点 C_1 の位置関係を表す定数 a, b を求めるためには、カメラはキャリブレーションが必要となる。そこで、校正を行わないかわりに、入力視点カメラの相対的な設置位置を工夫することを考える。具体的には、 C_1 が XY 平面上、 YZ 平面上、 ZX 平面上。あるいは X 軸上などに限定すると、式(4.12)、式(4.12)を定数 a, b を用いない方法で簡単化できる。以下、 XY 平面上、 YZ 平面上、 ZX 平面上の場合について詳しく説明する。

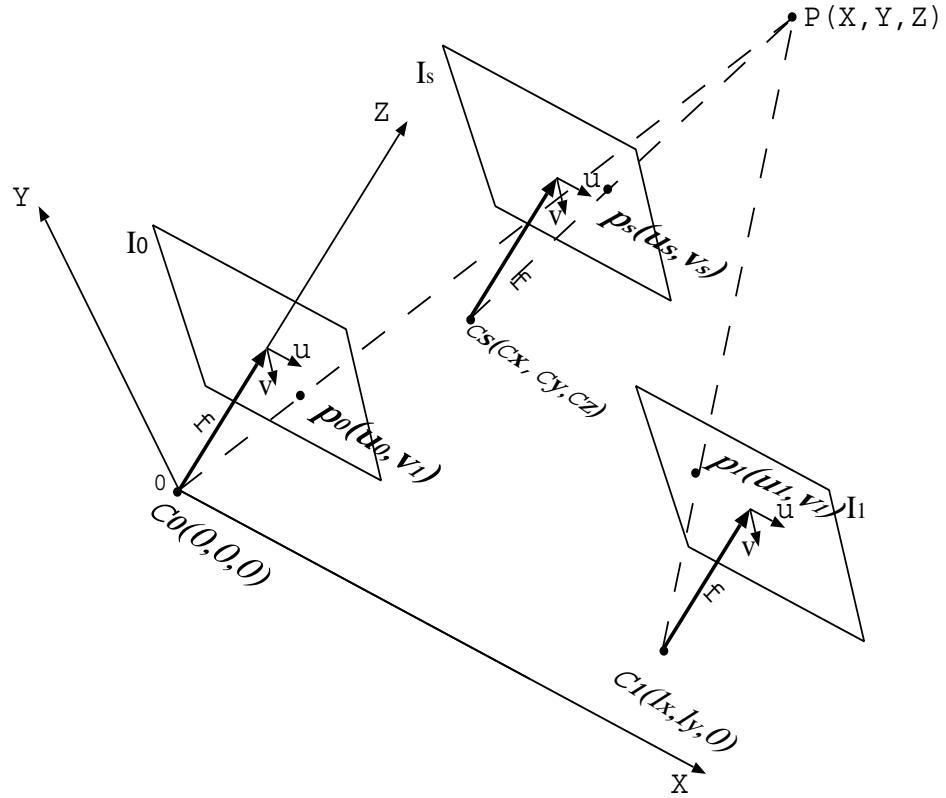


図 4.2：平行化した入力視点 XY 平面上の場合

例えば、平行化された視点およびその視点の画像平面位置は、図 4.2 のように置く場合、入力視点 C_0 、 C_1 それぞれ座標は $(0,0,0)$ 、 $(l_x, l_y, 0)$ となる。このとき、任意視点 $C_s(C_x, C_y, C_z)$ 対応の画像 I_s 上の対応点座標 (u_s, v_s) と、入力視点 C_0 、 C_1 それぞれ対応する画像 I_0 、 I_1 上の対応点座標 (u_0, v_0) 、 (u_1, v_1) の関係は式 (4.14) と式 (4.15) ようになる。

$$u_1 - u_0 \neq 0 \text{ 時 } \left\{ \begin{array}{l} u_s = \frac{u_0 l_x f + C_x f(u_1 - u_0)}{l_x f + C_z(u_1 - u_0)} = \frac{u_0 + \frac{C_x}{l_x}(u_1 - u_0)}{1 + \frac{C_z}{l_x f}(u_1 - u_0)}, \\ v_s = \frac{v_0 l_x f + C_y f(u_1 - u_0)}{l_x f + C_z(u_1 - u_0)} = \frac{v_0 + \frac{C_y}{l_x}(u_1 - u_0)}{1 + \frac{C_z}{l_x f}(u_1 - u_0)} \end{array} \right. \quad (4.14)$$

$$v_1 - v_0 \neq 0 \text{ 時} \left\{ \begin{array}{l} u_s = \frac{u_0 l_y f + C_x f (v_1 - v_0)}{l_y f + C_z (v_1 - v_0)} = \frac{u_0 \frac{l_y}{l_x} + \frac{C_x}{l_x} (v_1 - v_0)}{\frac{l_y}{l_x} + \frac{C_z}{l_x f} (v_1 - v_0)}, \\ v_s = \frac{v_0 l_y f + C_y f (v_1 - v_0)}{l_y f + C_z (v_1 - v_0)} = \frac{v_0 \frac{l_y}{l_x} + C_y f (v_1 - v_0)}{\frac{l_y}{l_x} + \frac{C_z}{l_x f} (v_1 - v_0)} \end{array} \right. \quad (4.15)$$

ここで、 $\alpha = \frac{C_x}{l_x}, \beta = \frac{C_y}{l_x}, \gamma = \frac{C_z}{l_x f}$ とすると、 $u_1 - u_0 \neq 0$ と $v_1 - v_0 \neq 0$ 時、任意

視点の画像上の対応点座標は、入力画像上の対応点座標から、式(4.14)と式(4.15)のどとらから算出できる。そして、式(4.14)は、任意視点位置変数 α, β, γ の設定だけによって、任意視点画像上の対応点座標を算出できる。更に、対応点の座標を式(4.15)に代入すると、定数 $\frac{l_y}{l_x}$ が求められる。式(4.15)も、任意視点位置変数 α, β, γ の設定だけによって、任意視点画像上の対応点座標を算出できる。ただし、入力視点 C_0 、 C_1 をX軸上の場合、 $v_1 - v_0 = 0$ である。

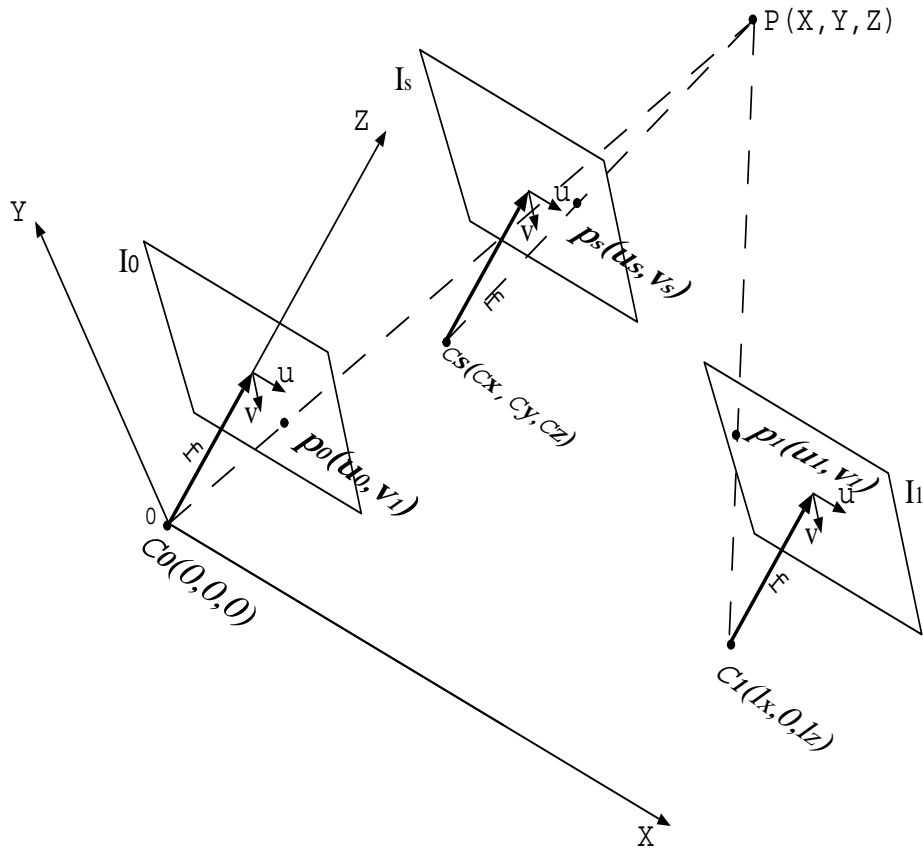


図 4.3 : 平行化した入力視点 XZ 平面上の場合

平行化した入力視点が図 4.3 のように XZ 平面上の場合は、入力視点 C_0 、 C_1 それぞれ座標は $(0,0,0)$ 、 $(l_x,0,l_z)$ となる。このとき、任意視点 $C_s(C_x, C_y, C_z)$ に対応する画像 I_s 上の対応点座標 (u_s, v_s) と、入力視点 C_0 、 C_1 それぞれに対応する画像 I_0 、 I_1 上の対応点座標 (u_0, v_0) 、 (u_1, v_1) の関係は

$$u_1 - u_0 \neq 0 \text{ 時} \left\{ \begin{array}{l} u_s = \frac{u_0 u_1 l_z - u_0 l_x f - C_x f(u_1 - u_0)}{l_x f + C_z(u_1 - u_0)} = \frac{u_0 u_1 - u_0 \frac{l_x f}{l_z} - \frac{C_x f}{l_z}(u_1 - u_0)}{\frac{l_x f}{l_z} + \frac{C_z}{l_z}(u_1 - u_0)}, \\ v_s = \frac{v_0 u_1 l_z - v_0 l_x f - C_y f(u_1 - u_0)}{l_x f + C_z(u_1 - u_0)} = \frac{v_0 u_1 - v_0 \frac{l_x f}{l_z} - \frac{C_y f}{l_z}(u_1 - u_0)}{\frac{l_x f}{l_z} + \frac{C_z}{l_z}(u_1 - u_0)} \end{array} \right. \quad (4.16)$$

$$v_1 - v_0 \neq 0 \text{ 時} \left\{ \begin{array}{l} u_s = \frac{u_0 v_1 l_z - C_x f(v_1 - v_0)}{-v_1 l_z + C_z(v_1 - v_0)} = \frac{u_0 v_1 - \frac{C_x f}{l_z}(v_1 - v_0)}{v_1 - \frac{C_z}{l_z}(v_1 - v_0)}, \\ v_s = \frac{v_0 v_1 l_z - C_y f(v_1 - v_0)}{v_1 l_z - C_z(v_1 - v_0)} = \frac{v_0 v_1 - \frac{C_y f}{l_z}(v_1 - v_0)}{v_1 - \frac{C_z}{l_z}(v_1 - v_0)} \end{array} \right. \quad (4.17)$$

である。

ここで、 $\alpha = \frac{C_x f}{l_z}$ 、 $\beta = \frac{C_y f}{l_z}$ 、 $\gamma = \frac{C_z}{l_z}$ とすると、 $u_1 - u_0 \neq 0$ と $v_1 - v_0 \neq 0$ 時、任意視点の画像上の対応点座標は、入力画像上の対応点座標から、式(4.16)と式(4.17)のどちらから算出できる。そして、式(4.17)は、任意視点位置変数 α, β, γ の設定だけによって、任意視点画像上の対応点座標を算出できる。更に、対応点の座標を式(4.16)に代入すると、定数 $\frac{l_x f}{l_z}$ が求められると、式(4.16)は、任意視点位置変数 α, β, γ の設定だけによって、任意視点画像上の対応点座標を算出できる。ただし、入力視点 C_0 、 C_1 が X 軸上にある場合、 $v_1 - v_0 = 0$ である。

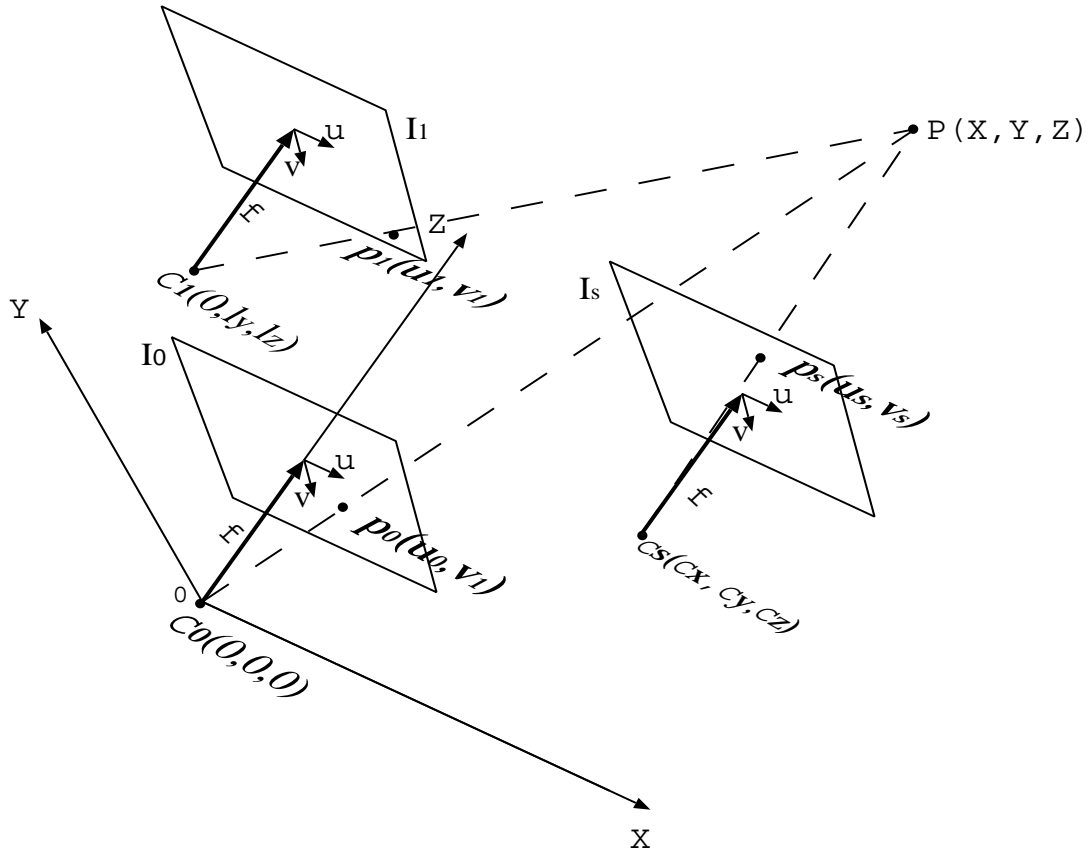


図 4.4：平行化した入力視点 YZ 平面上の場合

平行化した入力視点が図 4.4 のように YZ 平面上の場合は、入力視点 C_0 、 C_1 それぞれ座標は $(0,0,0)$ 、 $(0, l_y, l_z)$ となる。このとき、任意視点 $C_s(C_x, C_y, C_z)$ 対応の画像 I_s 上の対応点座標 (u_s, v_s) と、入力視点 C_0 、 C_1 それぞれ対応する画像 I_0 、 I_1 上の対応点座標 (u_0, v_0) 、 (u_1, v_1) の関係は

$$u_1 - u_0 \neq 0 \text{ 時} \left\{ \begin{array}{l} u_s = \frac{u_0 u_1 l_z - C_x f (u_1 - u_0)}{u_1 l_z - C_z (u_1 - u_0)} = \frac{u_0 u_1 - \frac{C_x f}{l_z} (u_1 - u_0)}{u_1 l_z - \frac{C_z}{l_z} (u_1 - u_0)}, \\ v_s = \frac{v_0 u_1 l_z - C_y f (u_1 - u_0)}{u_1 l_z - C_z (u_1 - u_0)} = \frac{v_0 u_1 l_z - \frac{C_y f}{l_z} (u_1 - u_0)}{u_1 l_z - \frac{C_z}{l_z} (u_1 - u_0)} \end{array} \right. \quad (4.18)$$

$$v_1 - v_0 \neq 0 \text{ 時} \left\{ \begin{array}{l} u_s = \frac{u_0 v_1 l_z - u_0 l_y f - C_x f (v_1 - v_0)}{-v_1 l_z + l_y f + C_z (v_1 - v_0)} = \frac{u_0 v_1 - u_0 \frac{l_y f}{l_z} - \frac{C_x f}{l_z} (v_1 - v_0)}{v_1 - \frac{l_y f}{l_z} - \frac{C_z}{l_z} (v_1 - v_0)}, \\ v_s = \frac{v_0 v_1 l_z - v_0 l_y f - C_y f (v_1 - v_0)}{v_1 l_z - l_y f - C_z (v_1 - v_0)} = \frac{u_0 v_1 - u_0 \frac{l_y f}{l_z} - \frac{C_x f}{l_z} (v_1 - v_0)}{v_1 - \frac{l_y f}{l_z} - \frac{C_z}{l_z} (v_1 - v_0)} \end{array} \right. \quad (4.19)$$

である。

ここで、 $\alpha = \frac{C_x f}{l_z}, \beta = \frac{C_y f}{l_z}, \gamma = \frac{C_z}{l_z}$ とすると、 $u_1 - u_0 \neq 0$ と $v_1 - v_0 \neq 0$ 時、任意

視点の画像上の対応点座標は、入力画像上の対応点座標から、式(4.18)と式(4.19)のどちらから利用できる。そして、式(4.18)は、任意視点位置変数 α, β, γ だけ、任意視点画像上の対応点座標を算出できる。更に、対応点の座標を式(4.19)に代入する

と、定数 $\frac{l_y f}{l_z}$ が求められる。式(4.19)も、任意視点位置変数 α, β, γ だけ、任意視点

画像上の対応点座標を算出できる。ただし、入力視点 C_0, C_1 が Y 軸上にある場合、 $u_1 - u_0 = 0$ である。

図 4.5 のような Seitz らの View Morphing と同じ視点位置関係の場合、入力視点 C_0, C_1 はそれぞれ座標は $(0,0,0), (l_x, 0, 0)$ となる。このとき、任意視点 C_s 対応の画像 I_s 上の対応点座標 (u_s, v_s) と、入力視点 C_0, C_1 それぞれ対応する画像 I_0, I_1 上の対応点座標 $(u_0, v_0), (u_1, v_1)$ の関係は

$$\left\{ \begin{array}{l} u_s = \frac{u_0 l_x f + C_x f (u_1 - u_0)}{l_x f + C_z (u_1 - u_0)} = \frac{u_0 + \frac{C_x}{l_x} (u_1 - u_0)}{1 + \frac{C_z}{l_x f} (u_1 - u_0)}, \\ v_s = \frac{v_0 l_x f + C_y f (u_1 - u_0)}{l_x f + C_z (u_1 - u_0)} = \frac{v_0 + \frac{C_y}{l_x} (u_1 - u_0)}{1 + \frac{C_z}{l_x f} (u_1 - u_0)} \end{array} \right. \quad (4.20)$$

のようになる。ただし、この場合、 $v_1 = v_0$ である。

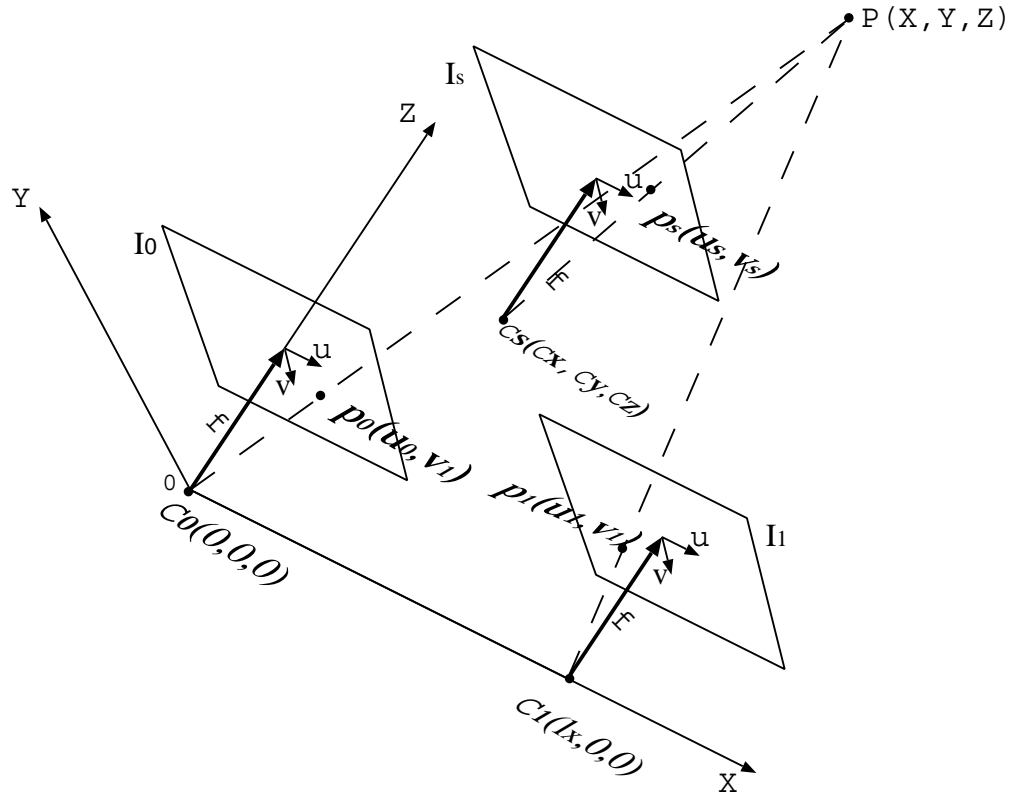


図 4.5：平行化した入力視点 X 軸上の場合

ここで、 $\alpha = \frac{C_x}{l_x}, \beta = \frac{C_y}{l_x}, \gamma = \frac{C_z}{l_x f}$ とすると、

$$\begin{cases} u_s = \frac{u_0 + \alpha(u_1 - u_0)}{1 + \gamma(u_1 - u_0)}, \\ v_s = \frac{v_0 + \beta(u_1 - u_0)}{1 + \gamma(u_1 - u_0)} \end{cases} \quad (4.21)$$

のように、 (α, β, γ) は任意視点位置を 1 つ定めるパラメータと見なすことができる。 α 、 β は l_x を単位長とした任意視点位置の X 、 Y 軸方向の変位量、 γ は $l_x f$ を単位量とした Z 軸方向の変位量である。それぞれの任意視点位置と (α, β, γ) は 1 対 1 に対応するので、入力視点間距離 l_x と焦点距離 f が分からなくても最終的な任意視点画像を生成できる。 α, β, γ の正負はそれぞれ、入力 2 視点より左右、上下、前後の視点位置と対応する。 $\gamma = 0$ のときは C_s の位置は XY 平面上であり、 $\beta = \gamma = 0$ ときは Seitz らの View Morphing と同じ状況設定となる。

4.4 本手法の適用条件

本手法では、2つの入力視点画像上の対応点から任意視点画像上の対応点を算出するが、任意視点画像のための視野設定も同じ内部パラメータを持つカメラを仮定している。このため、生成可能な任意視点画像の視野は入力画像を撮影するカメラのパラメータに依存する。また、本手法では、2つの入力視点画像から可視な部分进行处理の対象としている。このため、入力視点から撮影できない部分や、任意視点位置の設定によって同士の隠蔽が生じる部分は正しく処理できない。

4.5 シミュレーション実験と考察

序論でカメラのキャリブレーションが難しいことを説明した。もし、高精度の平行化ができれば、前節の図 4.5 のような平行化された入力視点が X 軸上の場合、 α, β, γ によらず、任意視点画像上の対応点座標は正確に求められる。しかし実際には、カメラを完全に平行に設置できないことなどによる誤差が生じ、任意視点画像上の対応点座標が本来あるべき座標からずれる。そこで以下では、理想的なカメラ設置からのずれが生じた場合に、任意視点画像上での対応点座標のずれの程度と、あるずれ量を定めた場合の α, β, γ の許容範囲をシミュレーション実験により検討した。

具体的には、カメラ C_1 はカメラ C_0 から l_x 離れるとし、焦点距離は $f = 400$ 画素で、 $(0.5l_x, 0.0, 3.0l_x)$ にある点を撮影するとする。カメラ C_1 が理想的な設置から平行移動と回転によって微小なずれを含む場合に、生成した任意視点画像と理想な任意視点画像を比べたときの対応点が何画素ずれるかによって、 α, β, γ の範囲への影響を調べる。

入力視点が X 軸上にある場合、カメラ間距離 l_x を単位長として考えるので、 X 軸方向の平行移動は対応点のずれに影響しない。さらに、カメラ設置の際にカメラの光軸回りの回転、すなわち、 Z 軸回りの回転は防ぎやすいと考えられる。そこでここでは、 Y 軸、 Z 軸に沿う方向の平行移動 $\Delta Y_T, \Delta Z_T$ と、 X 軸、 Y 軸回りの回転 $\Delta X_R, \Delta Y_R$ の4つを考慮することとする。

以下の議論では、理想的なカメラ設置での任意視点画像上の対応点座標と、ずれを含むカメラ設置位置での対応点座標との距離を誤差 e [画素] とし、 α, β, γ に対する e の分布を調べる。

4.5.1 各軸方向の平行移動の影響

カメラ C_1 が Y 軸方向にのみ平行移動した場合、 α, β, γ の範囲には影響を与えない。これは、 u_1 は変わらず、さらに、 $v_1 = v_0$ を仮定するため、 C_1 のカメラのずれの影響が無視されるためである。

次に、 C_1 が Z 軸方向にのみ平行移動する場合を考える。平衡移動 $\Delta Z_T = +0.1l_z$ での e の分布を図 4.6 に示す。上下の図は、それぞれ $\alpha - \beta$ 平面 ($\gamma = 0$) と $\alpha - \gamma$ 平面 ($\beta = 0$) での e の分布の等高線表示である。 $\gamma = 0$ のとき、 $\alpha < |0.4|, \beta < |0.4|$ で $e < 1$ であり、 γ が小さくなると、より広い α, β の範囲で誤差 e が 1 画素以内となる。図 4.6 の下図の右側につけた目盛は $f = 400$ [画素] のときの l_x を単位長とした値である。 ΔZ_T が小さくなる場合は、同程度の誤差を生じる α, β, γ の範囲はより広くなる。実際のカメラ設置では $\Delta Z_T < |0.01l_z|$ と考えて差し支えないので、 ΔZ_T は結果に対して大きく影響を与えないことがわかる。

4.5.2 各軸回りの回転の影響

まず、 X 軸回りの回転のみの場合を考える。回転量が $\Delta X_R = +10^\circ$ (左螺の向きを正) での e の分布を図 4.7 に示す。 $\gamma = 0$ のとき、 $\alpha < |1|, \beta < |1|$ で $e < 1$ であり、 γ が小さくなると、より広い α, β の範囲で誤差 e が 1 画素以内となる。

次に Y 軸回りの回転のみの場合を考える。回転量が $\Delta Y_R = +0.5^\circ$ での e の分布を図 4.8 に示す。 $\gamma = 0$ のとき、 $\alpha < |0.3|, \beta < |0.3|$ で $e < 1$ であり、 γ が小さくなると、より広い α, β の範囲で誤差 e が 1 画素以内となる。

回転量が小さくなると、同程度の誤差を生じる α, β, γ の範囲はより広くなる。実際のカメラ設置での回転量は 0.5° 以下と考えられ、微小な X 軸回りの回転は α, β, γ の範囲への影響は非常に弱い。しかし、 Y 軸回りの回転は X 軸回りの回転に比べて、 α, β, γ の範囲への影響は強いといえる。

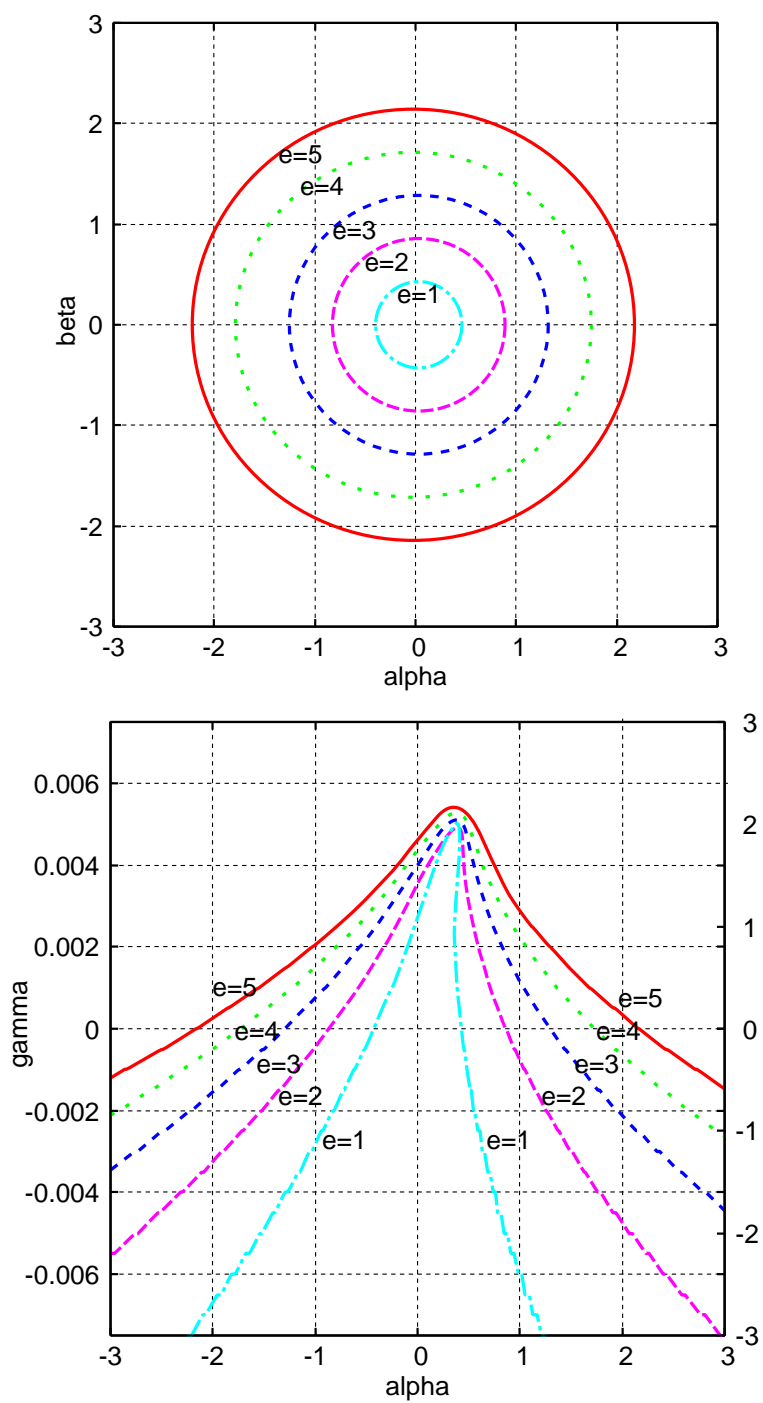


図 4.6: Z 軸方向の平行移動による誤差分布 ($\Delta Z_T = +0.1l_x$)

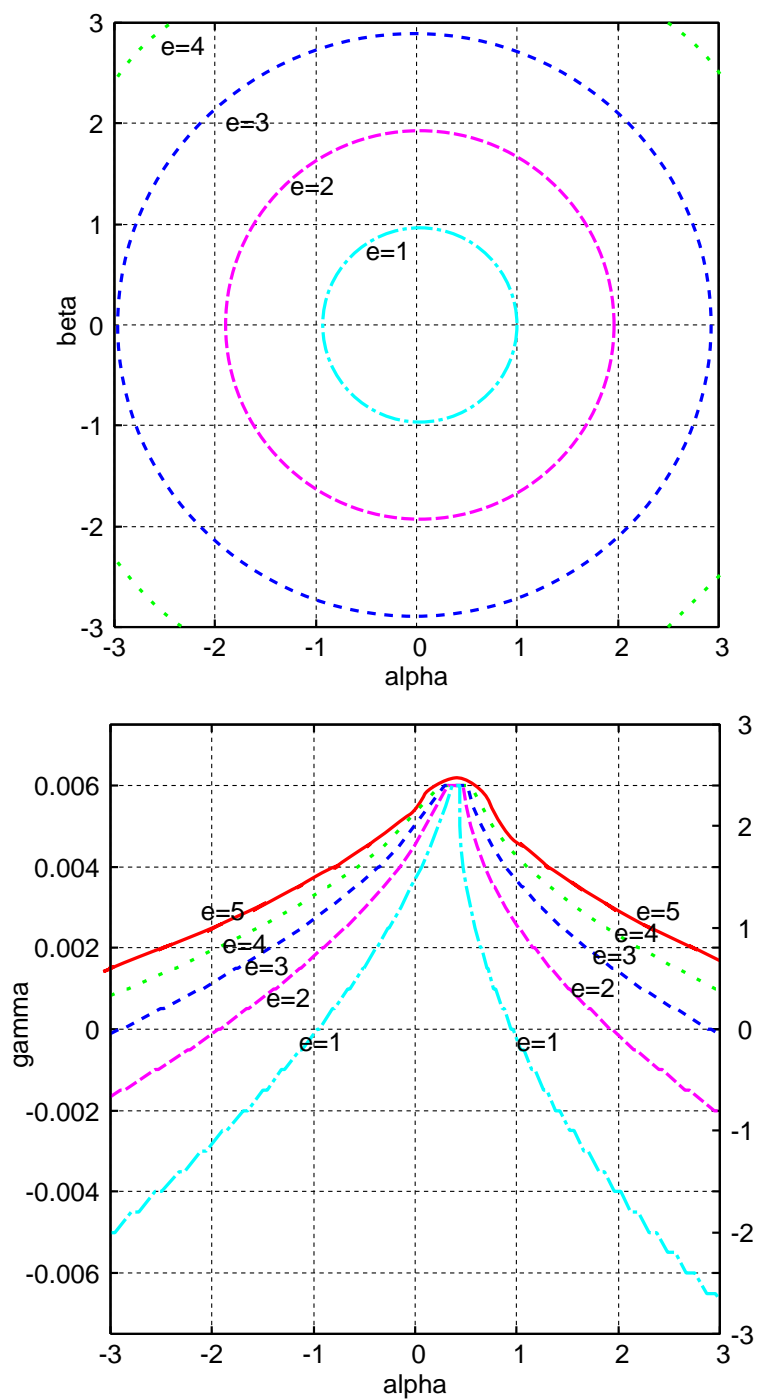


図 4.7: X 軸回りの回転による誤差分布 ($\Delta X_R = +10^\circ$)

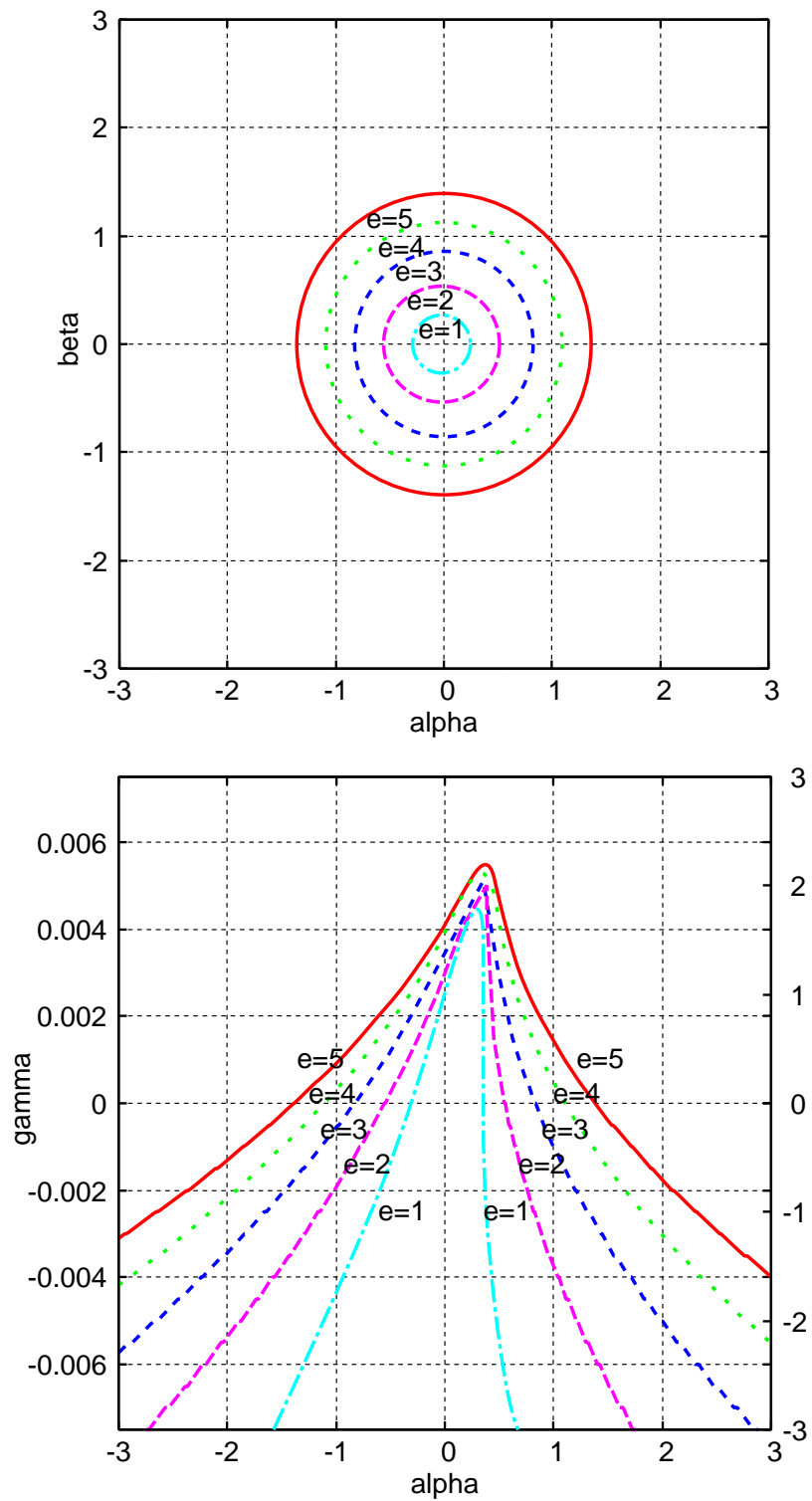


図 4. 8: Y 軸回りの回転による誤差分布 ($\Delta Y_R = +0.5^\circ$)

4.5.3 複数のずれの組み合わせによる影響

まず、平行移動が組み合わせられる場合として、カメラ C_1 が Y 軸と Z 軸方向にのみ平行移動した場合では、 e の分布は同じになる。これは、前者では、 Z 軸方向の平行移動分だけ u_1 が変化するが、式 (4.19) で $v_1 = v_0$ と仮定するため、 v_1 の変化は結果に影響せず、後者と同じになるからである。

次に、カメラ C_1 が回転量 $\Delta X_R = +10^\circ$ 、 $\Delta Y_R = +0.5^\circ$ の場合の e の分布を図 4.9 に示す。 $\gamma = 0$ のとき、 $\alpha < |0.2|$, $\beta < |0.2|$ で $e < 1$ であり、図 4.7 や図 4.8 と比べると、 X 軸と Y 軸の回転の組み合わせでは、 X 軸のみ、 Y 軸のみ回転の場合より α, β, γ の範囲は狭くなる。

さらに、回転と平行移動を組み合わせる場合を考察する Y 軸のみ平行移動した場合、 α, β, γ の範囲に全く影響はないが、図 4.10 に示すように、 Y 軸方向の平行移動 $\Delta Y_T = +0.1l_x$ と X 軸方向に対する回転 $\Delta X_R = +10^\circ$ を組み合わせる場合、 $\gamma = 0$ のとき、 $\alpha < |0.7|$, $\beta < |0.7|$ で $e < 1$ である。図 4.10 と図 4.7 と比べると、 Y 軸の平行移動と X 軸の回転と組み合わせるとき、 α, β, γ の範囲へ影響を与えることがわかった。

ただし、複数の視点ずれが組み合わせられた場合に誤差が小さくなる場合も確認された。例えば、 $\Delta Y_T = +0.1l_x$, $\Delta Z_T = +0.1l_x$, $\Delta X_R = +10$, $\Delta Y_R = -0.5$ などである、視点ずれによる投影点の変移量を $(\Delta u_1, \Delta v_1)$ とし、式 (4.19) より、 Δv_1 を無視して Δu_1 のみを考えると、このシミュレーションの状況設定では、それぞれの視点ずれが個別に生じた場合の変移量 Δu_1 は、 ΔY_T , ΔZ_T , ΔX_R , ΔY_R の順に、それぞれ、0.00, -2.30, -1.03, 3.58 画素である。これらの複数の視点ずれを組み合わせた場合の総合的な変移量の総和となり、その値が 0 に近い場合は個々のずれの影響が相殺される。実際、上記位置ずれ設定では Δu_1 の総合は 0.25 画素であり、誤差分布 (図 4.14) を見ると、個別のずれが生じた場合 (図 4.6, 図 4.7, 図 4.8) に比べて誤差が小さくなっていることがわかる。

本提案の方法では、カメラ設置のずれとして、 Y 軸方向の平行移動のときずれが生じず、 Z 軸方向の平行移動と X 軸回りの回転では、 α, β, γ の範囲への影響は非常に弱いことが検証できた。しかし、 Y 軸回りの回転は、ほかのずれより α, β, γ の範囲への影響が強いことがわかった。しかし実際のカメラ設置 (平行化方法) で Y 軸回りの回転が 0.5° 以下にできると考えられ、 α, β, γ に関しては十分広い範囲で、誤差の少ない任意視点画像を生成できるといえる。

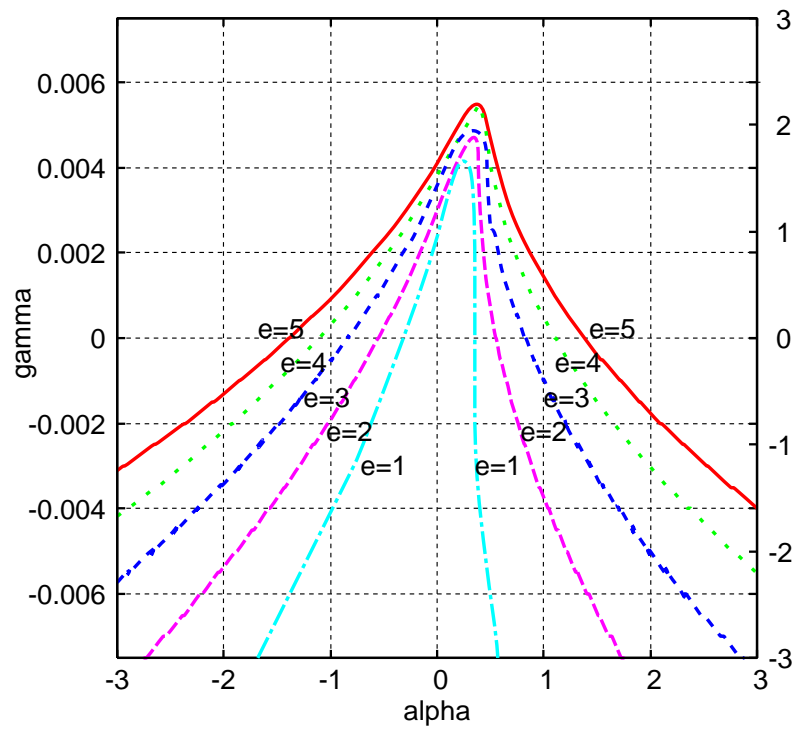
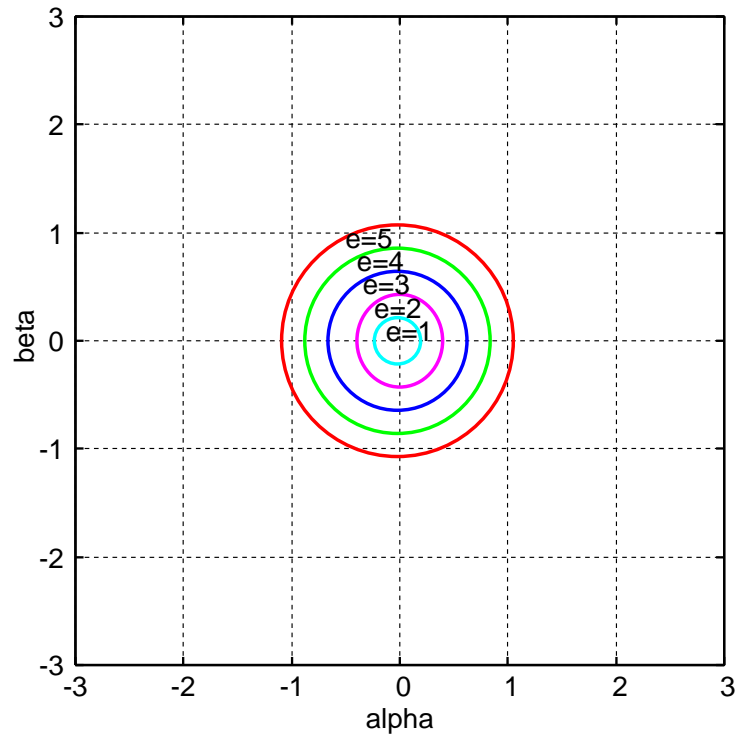


図 4.9: 2 軸の回転の組み合わせによる誤差分布 ($\Delta X_R = +10^\circ$ $\Delta Y_R = +0.5^\circ$)

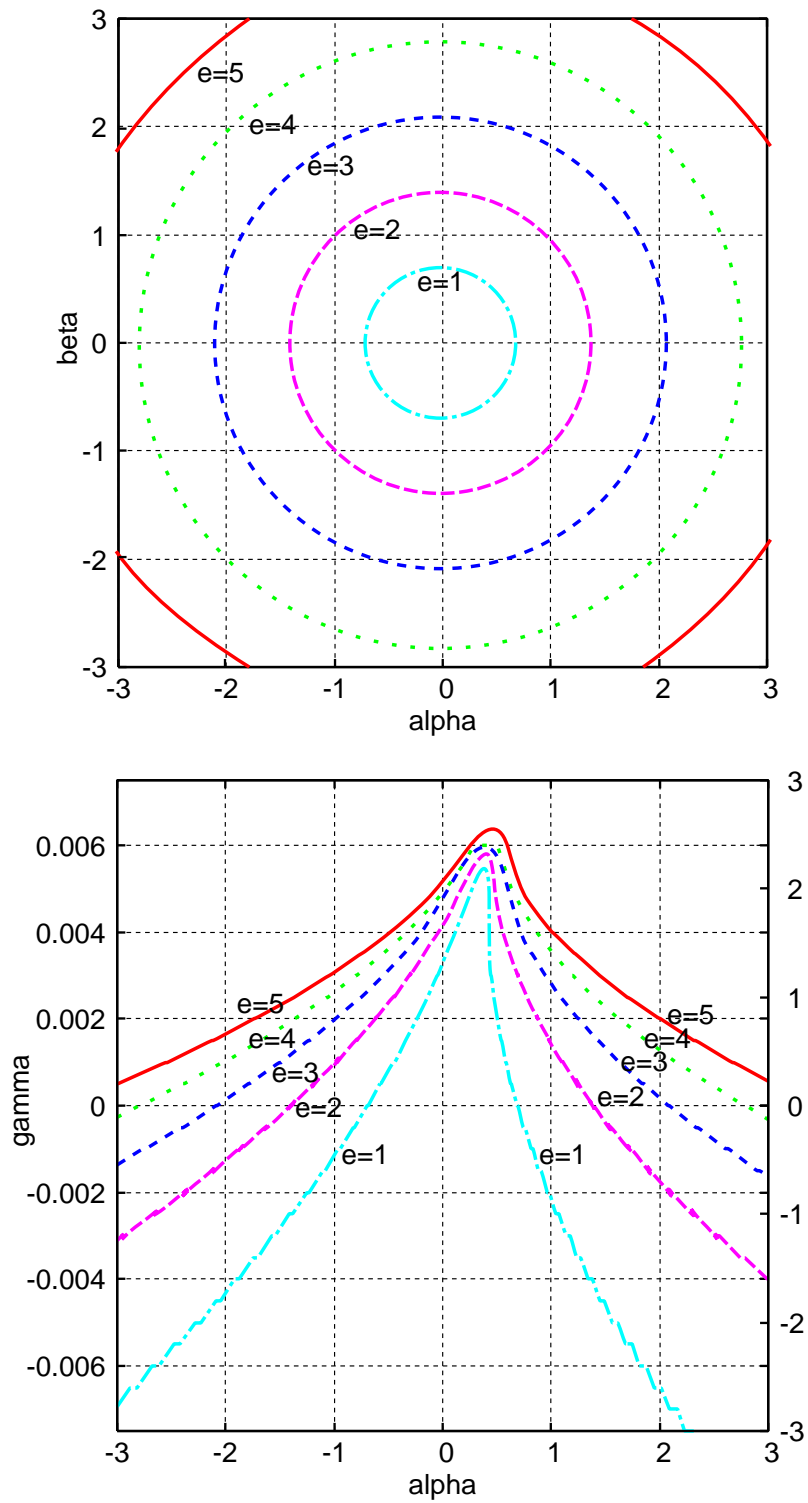


図 4.10：平行移動と回転の組合せによる誤差分布 ($\Delta Y_T = +0.1 \text{ } l_x$, $\Delta X_R = +10^\circ$)

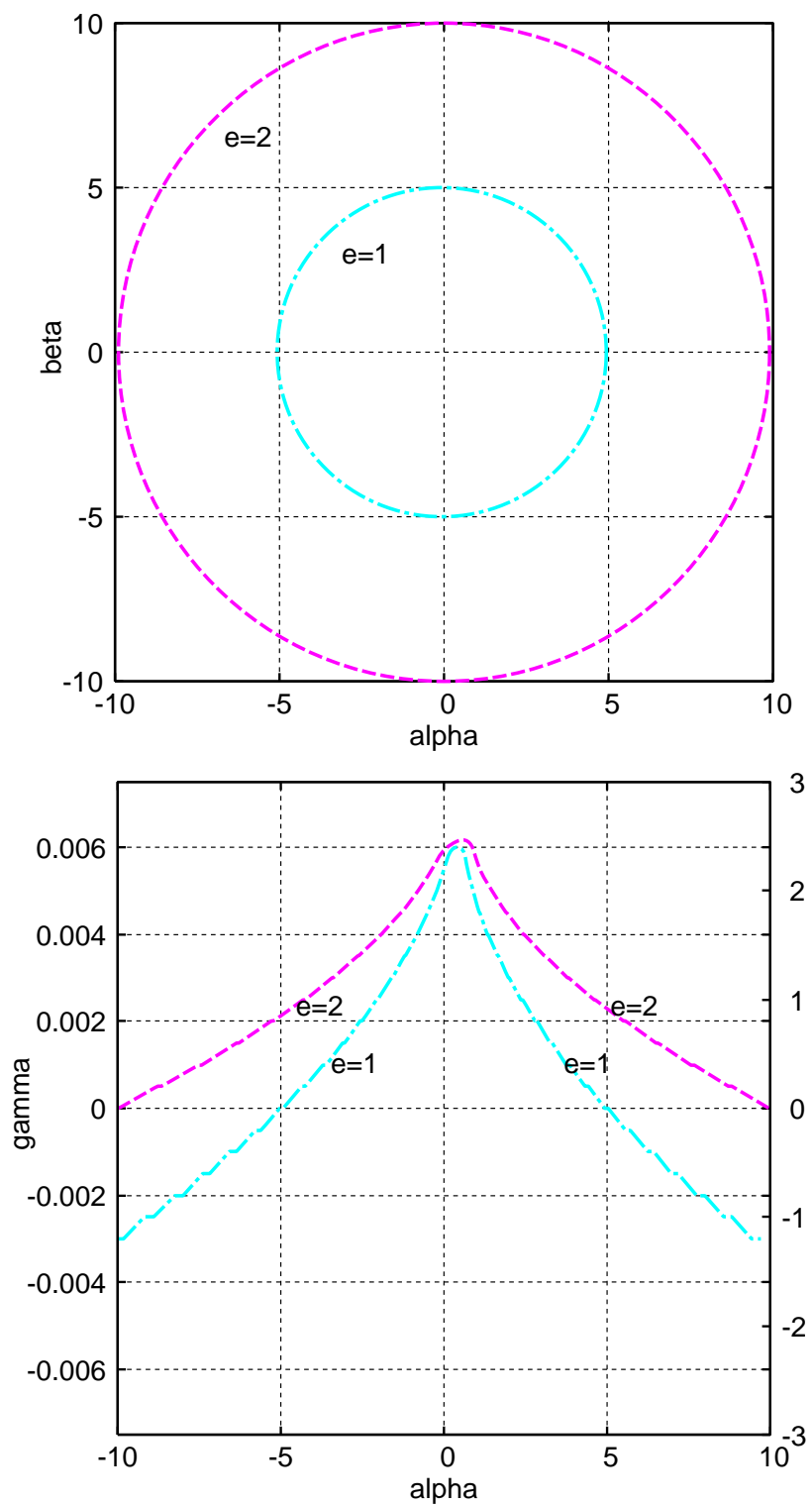


図 4.11: 複数のずれの組合せで α, β, γ の範囲が広がる例
 $(\Delta Y_T = +0.1l_x, \Delta Z_T = +0.1l_x, \Delta X_R = +10^\circ, \Delta Y_R = -0.5^\circ)$

第五章 任意視点画像の生成

前章には、射影幾何に基づく任意視点画像上の対応点と入力視点画像上の対応点関係を導き、そして、その関係の有効性の実験と考察を行った。本章では、前章の入力視点と入力視点間の関係を用いて、実撮影環境で実験を行って、そして、合成した任意視点画像を用いて、本文提案の手法の有効性を確認する。

5.1 任意視点画像生成の流れ

View Morphing 処理のような一般的の任意視点画像生成の流れは、まず、入力視点画像上の対応点の群を各々の画像で指定する。次は、平行化入力画像を生成する。それから、三角形パッチを生成して、平行化任意視点画像上の対応点座標を求めて、テクスチャマッピングを貼り付ける。最後に、任意視点画像を作る。

5.2 具体的処理

今回の実環境で考察実験では一般的任意視点画像生成の流れを用いて、任意視点画像生成する。以下で任意視点画像生成の流れについて述べる[6]。

5.2.1 対応点の指定

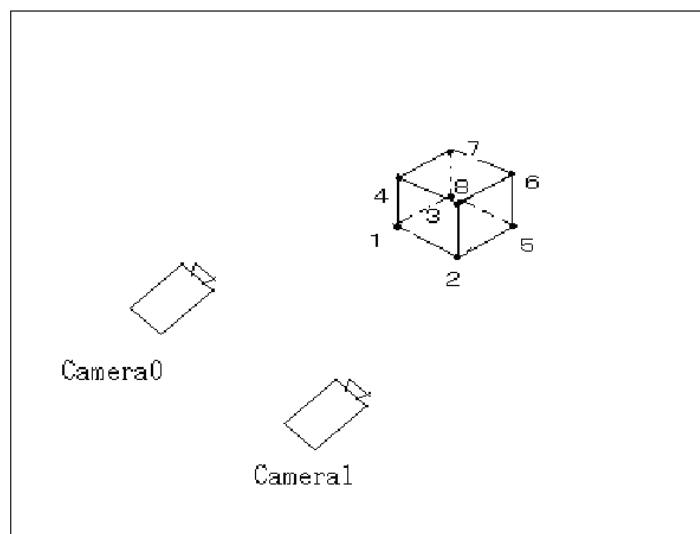


図 5.1 : 立方体

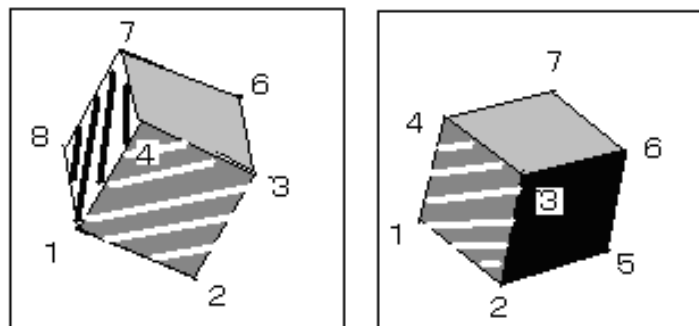


図 5.2：立方体の対応点の指定

まず、任意視点画像 I_s を 2 つ入力視点画像 I_0, I_1 における点の対応関係から生成するために、入力視点画像 I_0, I_1 から対応をとりながら N 個の対応点を手動で指定する。そこで、2 つのカメラで立方体を撮影し(図 5.1)、得られた 2 つの視点画像から対応点を指定した例を図 5.2 に示す。

このとき、各々の視点画像に対応点が写っており、左のカメラは対応点 5 だけ写っていない、右のカメラは対応点 8 だけが写っていない。

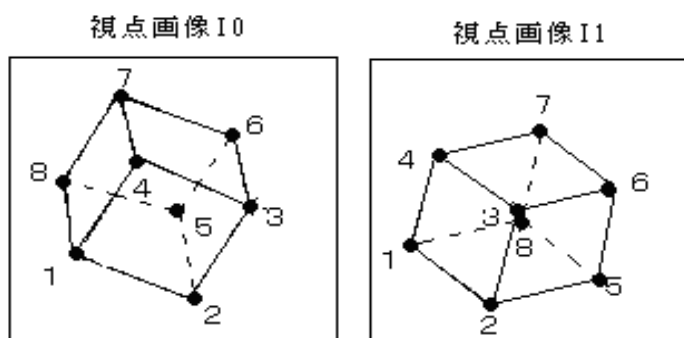


図 5.3：見えない対応点の取得の例

図 5.2 ように、一方 1 つの視点画像で見えて、もう一方の視点画像で見えない対応点があると、その対応点で構成できる三角形パッチは生成されず、任意視点画像のその三角形パッチの部分は喪失されてしまうので、一方、見えない対応点を、予測で指定必要することがある。

そこで、ある視点画像で見えない対応点にも手動で予測して対応点を指定する。図 5.2 のように一方の視点画像では対応点 5 が見えなく、もう一方視点画像では対応点 8 が見えない。そこで、図 5.3 のように見えない対応点についてもその位置を予測し指定する。その視点画像ではこの対応点は見えないという可視情報を付ける。図 5.4 は入力視点画像上の対応点の指定の例である。

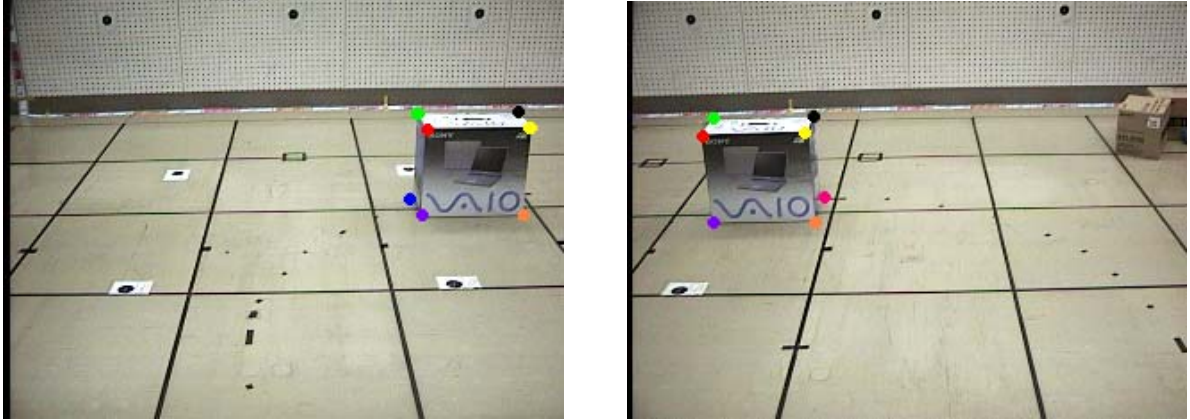


図 5.4 : 入力視点画像上の対応点の指定の例

5.2.2 平行化視点画像の生成

次に、入力視点画像 I_0, I_1 に対してパン・チルト・焦点距離 $(0, 0, f)$ へ透視投影変換を行い、2つの視点の光軸を平行となる平行化視点画像を生成する（図 5.5）。これは、幾何学関係を正確に反映させた任意視点画像の対応点の位置を求めるためである。そして、平行化入力視点画像から実3次元空間中の点 $P(X, Y, Z)$ に対する対応点 $p_0(u_0, v_0), p_1(u_1, v_1)$ を求める。また、他の対応点群についても平行化視点画像上での対応点座標を求める。

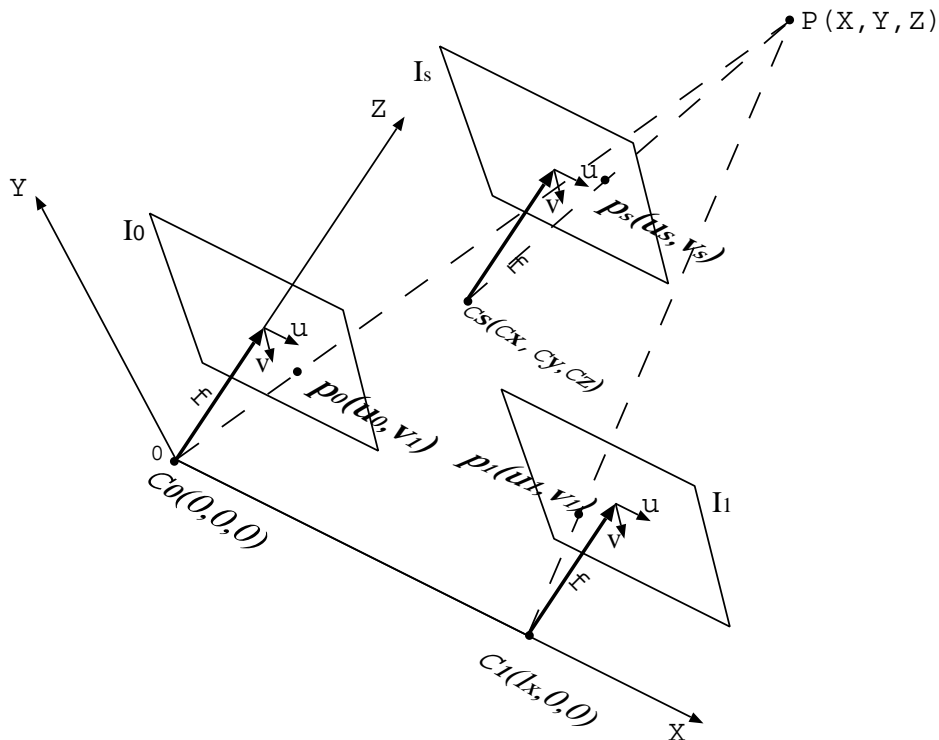


図 5.5: 平行化視点画像の生成

5.2.3 三角形パッチの生成

平行仮想視点画像の生成し、その画像上での対応点群を求めた[6]。そして、視点位置比率を用いた平行化任意視点画像上の対応点座標を算出することができる。しかし、限られた2次元点群データだけでは平行化任意視点画像を生成することができない。生成するためにはテクスチャを貼る三角形群が必要になる。そこで、まず、平行化視点画像の対応点群を接続し三角形群を生成する。そして、その三角形群を平行化任意視点画像の対応点群に用いて任意視点での三角形群を求める。図5.7は、入力視点画像の三角形分割した例である。

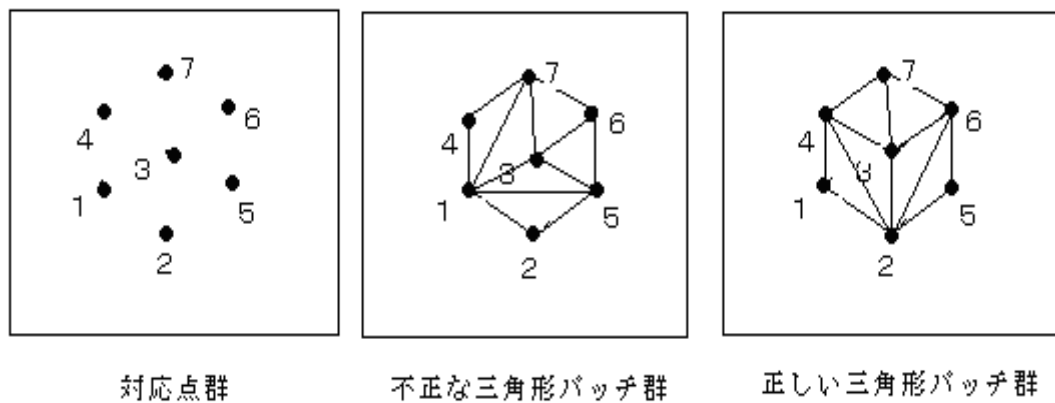


図 5.6：立方体における三角形分割

どちらの視点画像のテクスチャをはるのか知るためにも必要となる。そこで、三角形分割で得られる三角形パッチにタイプ判定を行う。これは、可視情報が付加した対応点を扱うので、三角形パッチにも可視情報の付加が必要になる。そして、この情報はテクスチャマッピングを行う上で、全ての三角形パッチを以下のように判定する。

- タイプ0: 全ての視点で見える三角形パッチ
- タイプ1: 視点0だけで見える三角形パッチ
- タイプ2: 視点1だけで見える三角形パッチ

そこで、三角形パッチのタイプ判定を行うために以下の2種類の情報を用いる。

- 三角形パッチを構成する3つの頂点の可視情報
- 三角形パッチの外積の符号情報

ある視点で、三角形パッチを構成する 3 つの頂点の中で 1 つでも不可視の対応点があれば、その三角形パッチはその視点からみえない。また、その条件を満たしていても、2 つの視点画像で各々計算した三角形パッチの外積の符号が異なる場合、三角形パッチが裏返った状態になり、可視、不可視が中間視点のどこかで切り替わることがわかる。もし、視点 0 で可視の三角形パッチが視点 1 での三角形パッチの外積符号と違うならば、視点 1 では不可視と判断できる。

表 5.1：三角形パッチのタイプ判定条件

順番	各視点の 3 頂点の見え方 ＊ 1		各視点での三角形パッチの外積符号＊ 2		タイプ
	視点 0	視点 1	視点 0	視点 1	
視点 0	1	1	－	－	0
	1	0	/	/	1
視点 1	1	1	正	負	2
			負	正	
	0	1	/	－	2

＊ 1．各視点で三角形パッチの 3 頂点の対応点が全て見えるのなら 1, そうでないのなら 0。

＊ 2．－は正負のどちらでもよい、/はその視点の三角形パッチの外積符号情報は使わない。

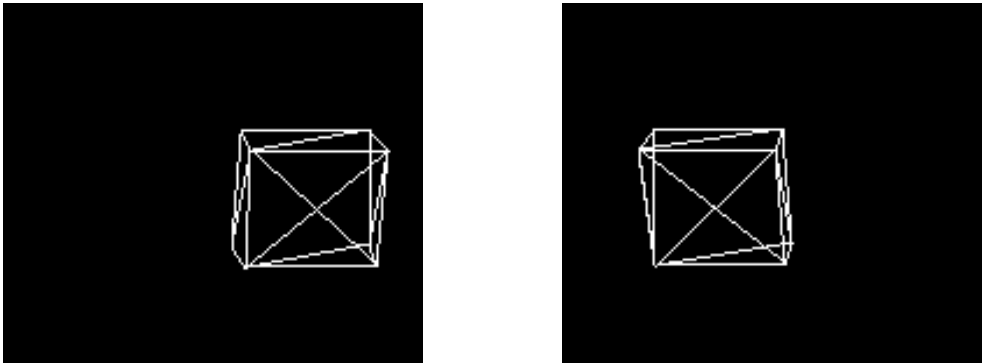


図 5.7：入力視点画像の三角形分割

5.2.4 平行化任意視点画像上の対応点座標の算出

これまでの処理により、入力視点画像の光軸が平行となる平行化視点画像 I_0, I_1 を生成し、 I_0, I_1 上の全ての対応点座標が求まった。次に、位置比率を用いた平行化任意視点画像 I_s 上のすべての対応点座標を求める（図 5.8）。実3次元空間中の点 $P(X, Y, Z)$ に対する平行化任意視点画像 I_s 上の対応点 $p_s(u_s, v_s)$ は、平行化視点画像 I_0, I_1 上の対応点 $p_0(u_0, v_0), p_1(u_1, v_1)$ を用いて以下の式で表される。

$$\begin{cases} u_s = \frac{u_0 + \alpha(u_1 - u_0)}{1 + \gamma(u_1 - u_0)}, \\ v_s = \frac{v_0 + \beta(u_1 - u_0)}{1 + \gamma(u_1 - u_0)} \end{cases} \quad (5.1)$$

式 (5.1) 中 α, β, γ とは、任意視点の位置比率という。

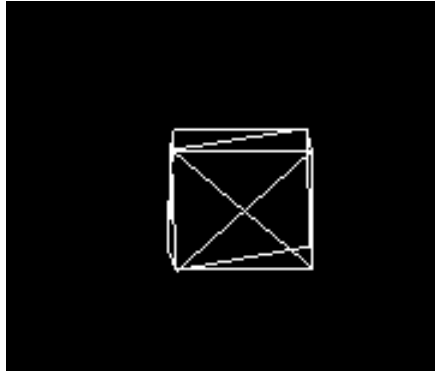


図 5.8：任意視点画像上の対応点と三角形パッチの算出

5.2.5 テクスチャマッピング

これから、平行化任意視点画像 I_s 上の対応点座標を算出し、平行化視点画像から三角形パッチ群を生成した。次に、その三角形パッチ群にテクスチャマッピングを行い、平行化任意視点画像 I_s を生成する。テクスチャマッピングとは、3次元コンピュータグラフィックスで物体の表面の質感を表現するために画像を貼り付けることである。そして、貼り付ける画像のことをテクスチャという。同じ立方体のグラフィックスでも、金属のテクスチャを貼り付ければ金属片に見え、木目のテクスチャを貼り付ければ木片に見える。

本研究では、2次元アフィン変換を用いてテクスチャマッピングを行う。2次元アフィン変換とは、拡大、縮小、位置移動、回転の幾何学変換のことである。ある点 (x, y) が拡大、縮小、位置移動、回転の幾何学変換で (X, Y) に位置を変えるとすると、つぎのような1次多項式、

$$X = e_0x + e_1y + e_2 \quad (5.2)$$

$$Y = e_3x + e_4y + e_5 \quad (5.3)$$

で示される。この $e_0 \sim e_5$ のアフィンパラメーターには拡大、縮小、位置移動、回転のパラメーターが全て含まれている。

テクスチャマッピングでは、三角形パッチの裏返し判定を行う。これは、裏返しになった三角形パッチにテクスチャを貼り付けないためである。テクスチャマッピングの概略図を図5.9に示す[6]。

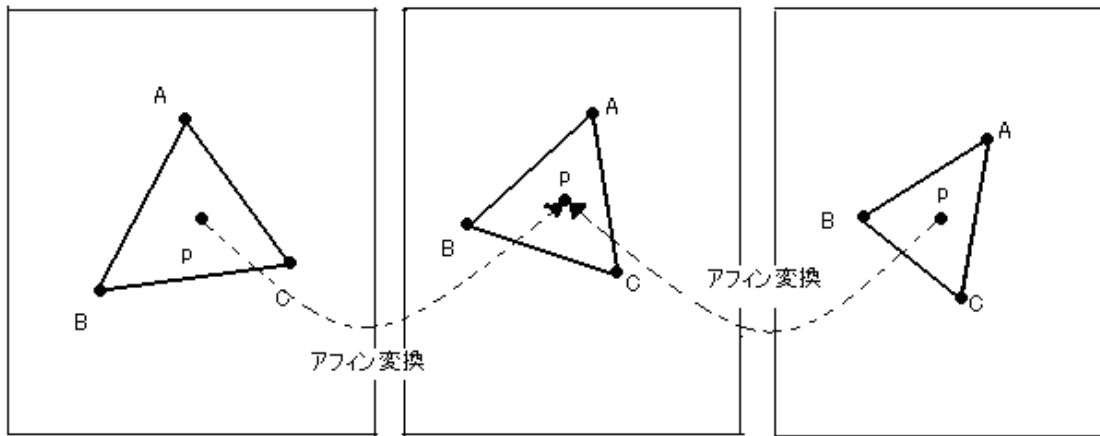


図 5.9 : テクスチャマッピングの概略図

5.2.6 最終的任意視点画像への変換

最後に、テクスチャマッピングから平行化任意視点画像を生成する。任意のパン・チルト・焦点距離より、平行化任意視点画像 I_p を任意視点のパン・チルト・焦点距離の仮想平面に逆透視投影を行って、任意のパン・チルト・焦点距離視点画像を生成することができる。図5.10は、生成した任意視点画像の例である。



図 5.10 : 生成した任意視点の画像

5.3 実験

本手法の有効性を確認するために比較的単純なシーン状況での実画像を用いた任意視点画像生成実験を行った。

今回の実験では、内部パラメータが同じ 2 台のビデオカメラ (Sony 社の EVI-G20) を図 5.11 のように配置した。カメラの焦点距離 $f = 400$ 画素、床からの高さ 50cm、カメラ間距離は 120cm とし、光軸と画像の軸がそれぞれ平行となるようにカメラを設置している。約 3 m 前方の箱を撮影した画像を入力画像とした。入力画像を図 5.12 に示す。入力画像サイズは 320×320 である。

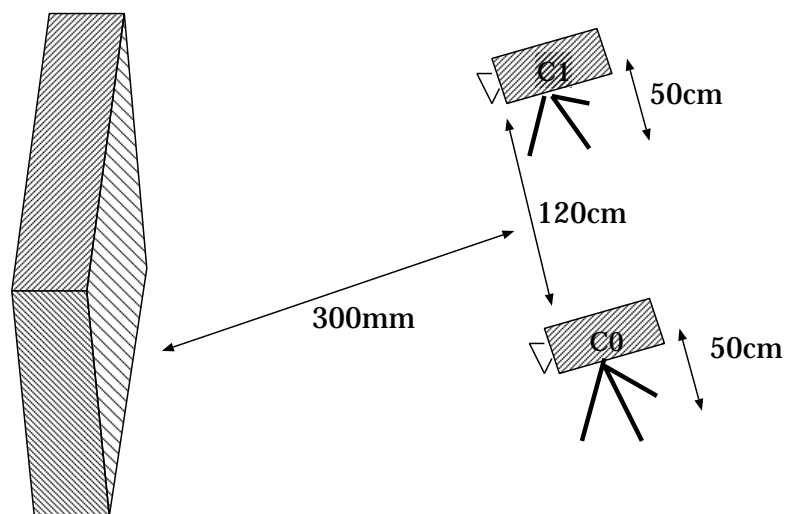


図 5.11 : 実験状況



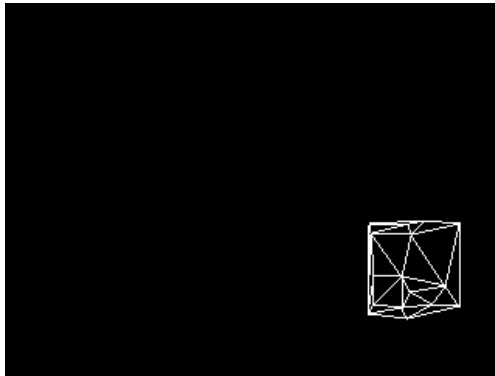
I_0 for camera C_0



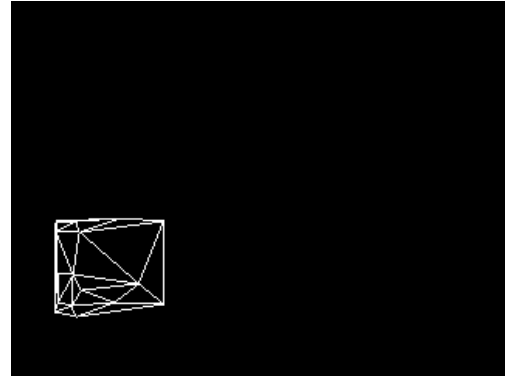
I_1 for camera C_1

図 5.12 : 入力視点画像

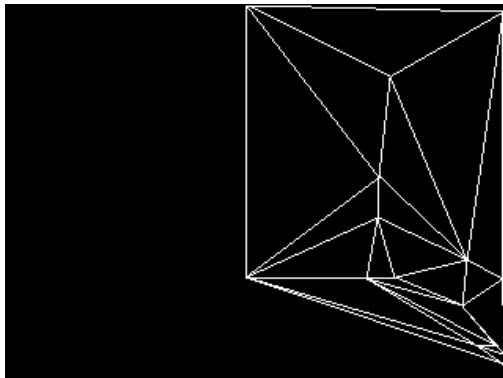
ここでは、背景と対象を分けて処理する。対象物には 16 個、背景には 15 個の対応点を与えると、入力画像での対象物と背景の三角形パッチを図 5.13 に示す。



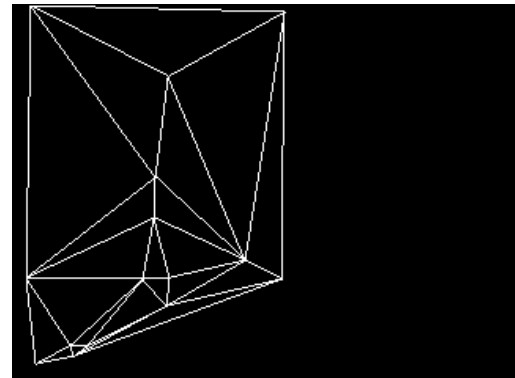
For object on I_0



For object on I_1



For background on I_0



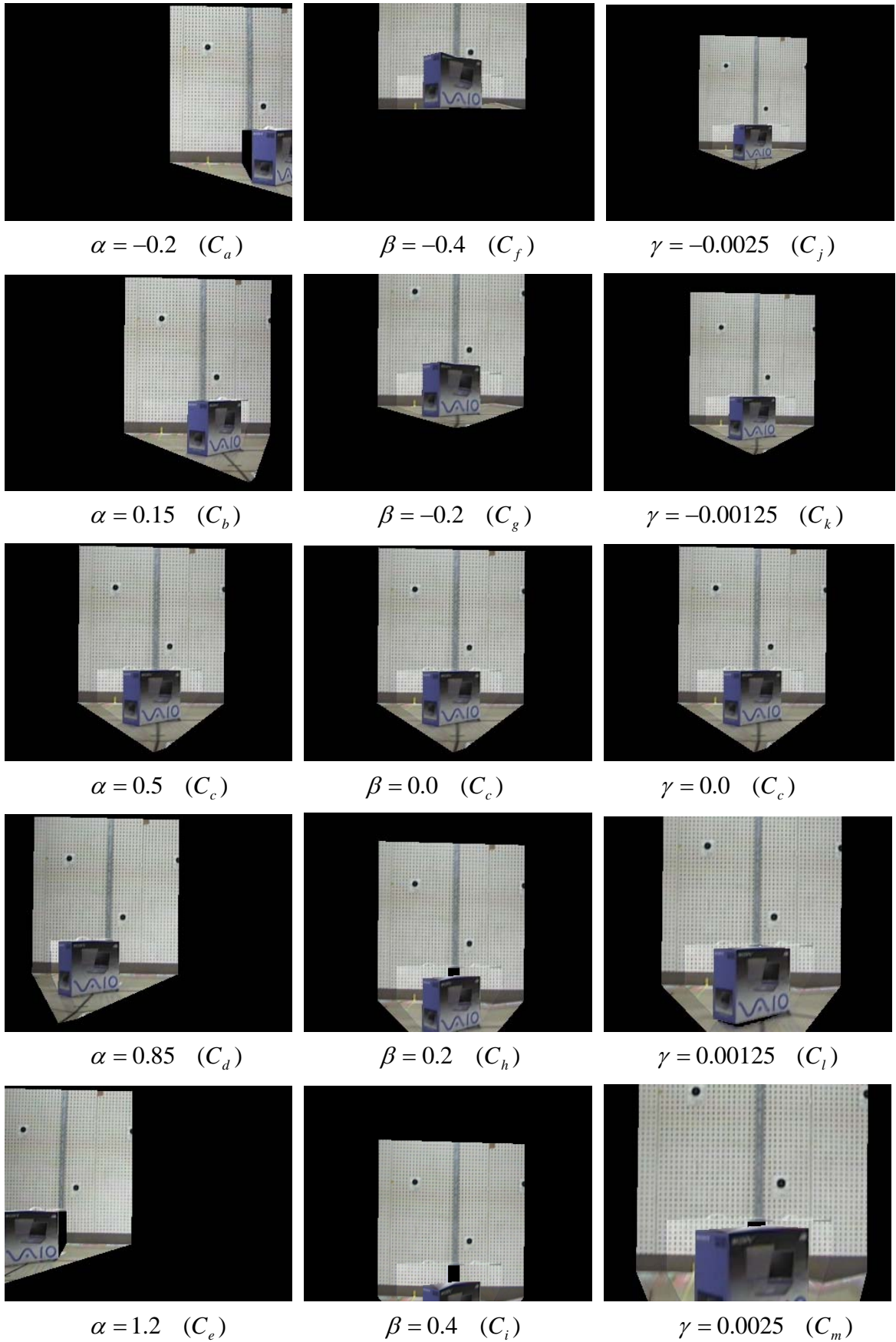
For background on I_1

図 5.13 : 入力視点画像上の対応点三角形パッチ

表 5.2 : 合成した任意視点画像位置

Synthesis View	(α, β, γ)	(C_x, C_y, C_z) [cm]
C_a	$(-0.2, 0, 0)$	$(-24, 0, 0)$
C_b	$(0.15, 0, 0)$	$(18, 0, 0)$
C_c	$(0.5, 0, 0)$	$(60, 0, 0)$
C_d	$(0.85, 0, 0)$	$(102, 0, 0)$
C_e	$(1.2, 0, 0)$	$(144, 0, 0)$
C_f	$(0.5, -0.4, 0)$	$(60, -48, 0)$
C_g	$(0.5, -0.2, 0)$	$(60, -24, 0)$
C_h	$(0.5, 0.2, 0)$	$(60, 24, 0)$
C_i	$(0.5, 0.4, 0)$	$(60, 48, 0)$
C_j	$(0.5, 0, -0.0025)$	$(60, 0, -120)$
C_k	$(0.5, 0, -0.00125)$	$(60, 0, -60)$
C_l	$(0.5, 0, 0.00125)$	$(60, 0, 60)$
C_m	$(0.5, 0, 0.0025)$	$(60, 0, 120)$
CA	$(0.75, 0, -0.00125)$	$(90, 0, -60)$
CB	$(0.5, -0.08330, 0.00125)$	$(60, -10, 60)$

表 5.2 に示す α, β, γ の組、すなわち、任意視点 (C_x, C_y, C_z) の設定に対する生成画像の例を図 5.14 に示す。図 5.14 の (a), (b), (c) は、それぞれ α, β, γ の変化に対応する。図 5.14 の (a) において、対象物である箱の左側面の見え方に注目すると、 X 軸に沿う視点の移動に伴う見え方の変化を正しく表していることが分かる。これは、Seitz らの手法と視点変化と同じである。次に、図 5.14 の (b) において、視点の Y 軸方向の移動に伴う見え方の変化を正しく表していることが分かる。さらに、図 5.14 の (c) において、箱と床の境界部分の見え方に注目すると、 Z 軸方向の視点移動に伴う奥行き表現の変化を正しく表していることがわかる。このように、 Y 軸、 Z 軸方向の視点移動に対しても幾何学的に正しい任意視点画像を生成できることが確認でき、本手法の視点設定の自由度が高いことを示している。



(a) α の変化 ($\beta=0.0, \gamma=0.0$) (b) β の変化 ($\alpha=0.5, \gamma=0.0$) (c) γ の変化 ($\alpha=0.5, \beta=0.0$)

図 5. 14 : 合成した任意視点画像

なお、任意視点位置を撮影視点から大きく変えた場合、生成画像の中に、二つの入力撮影視点のどちらからも観測できない部分が生じる。図 5.14 の視点 C_a, C_g, C_h, C_i, C_m の生成画像では、箱の横や上の部分にそのような領域が生じ、テクスチャが貼られていない黒い部分が残っている。この問題は、周囲の領域との連続性や物体形状の仮定を導入したテクスチャ推定を行うなどである程度軽減できると考えられる。

5.4 考察

図 5.15 は、入力画像の撮影位置とは異なる位置で実際に撮影した画像と、その撮影位置に対応する α, β, γ を用いて生成した画像を比較したものである。箱とその周囲の状況について実撮影画像とほぼ同じ生成画像を得ることができ、本手法の有効性を示している。

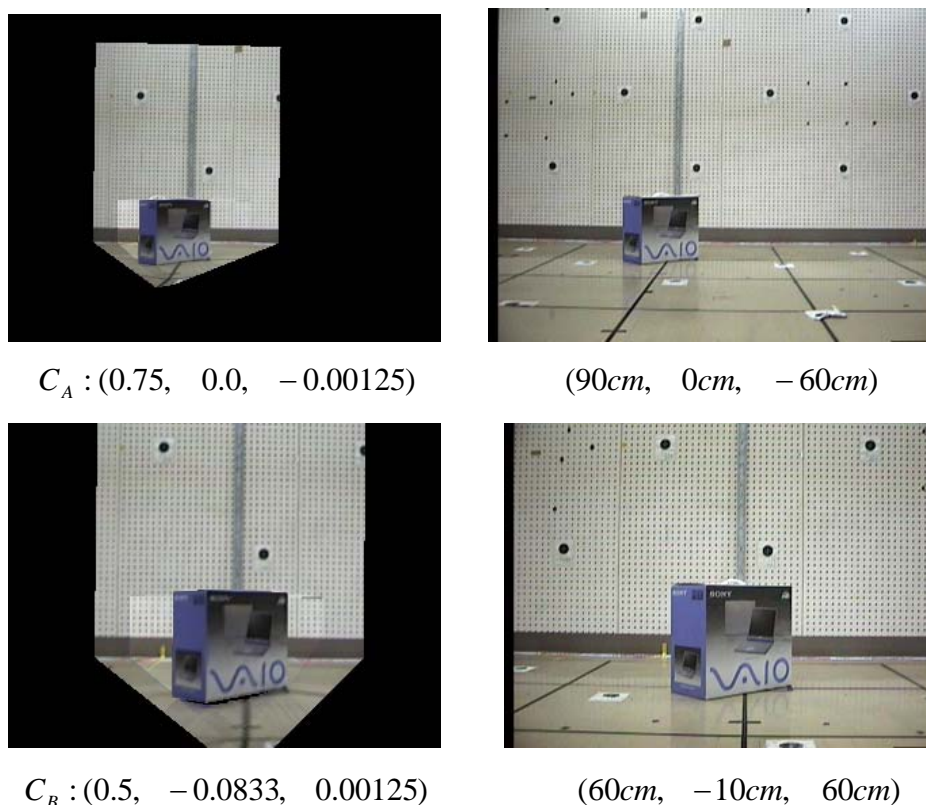


図 5.15：合成画像と実撮影画像の比較

第六章 まとめ

本文では、Seitz らが提案した View Morphing 手法を拡張し、少ないカメラ情報を利用しながら、入力視点画像から、空間中の任意視点画像を生成できる手法について述べた。この手法では、未校正の2つのカメラから得られた視点撮影画像から、3次元モデルは復元せずに任意視点画像が生成でき、更に、その視点位置設定も3次元空間中の任意の点として従来の方法より広範囲で設定できるものである。

実環境の実験より、この手法で生成した任意視点画像はほぼ良好な結果となることを確認した。更に、シミュレーション実験により、2台のカメラが完全に平行でない場合でも、三次元的に十分広い範囲で、誤差の少ない任意視点画像を生成できることを確認できた。

しかし、実撮影画像の照明環境と合成視点での照明環境が異なる場合や、カメラの設置、画像上の対応点指定などに精度の問題があり、合成した任意視点画像と同じ視点で撮影した実画像とは完全には一致しない。これらの解決は今後の課題である。また、広い空間で高品質の任意視点画像を生成するため、対応点の指定手法を検討する必要がある。

参考文献

- [1] Steven M. Seitz and Charles R. Dyer, “View Morphing”, Proceedings of SIGGRAPH’ 96, pp. 21-31, 1996.
- [2] Jiangjian Xiao and Mubarak Shah, “From Images to Video: View Morphing of Three Images”, Conference on Vision, Modeling and Visualization, pp. 495-502, November 19-21, Munich, Germany 2003.
- [3] 出口光一郎 著, “ロボットビジョンの基礎”, コロナ社, pp. 34-35, 2000.
- [4] 木村誠, 斎藤英雄, “3カメラ間の射影幾何を用いたView Morphing”, 映像情報メディア学会誌, Vol. 57, No. 10, pp. 1342-1346 (2003).
- [5] 北原格, 大田友一: “多視点映像の融合によるスポーツシーンの自由視点映像生成—3次元形状表現用平面の適応的配置”, PRMU2000-189, pp. 23-30 (2001).
- [6] 野村直恭, “多視点撮影画像に基づく任意視点画像生成に関する研究”, 平成14年度 福井大学 大学院工学研究科 情報工学専攻 修士論文 2003.
- [7] 藤井俊彰, “3次元統合画像符号化の基礎検討”, 東京大学大学院工学系研究科 博士論文 1995.
- [8] 村上伸一 著, “コンピュータグラフィックスの基礎”, 東京電機大学出版局 1998.

謝 辞

本研究を進めるにあたって御指導下さいました東海彰吾准教授、長谷博行教授に対し心から深く感謝致します。

また、本論文を御査読頂きました岡崎耕三教授、荻原隆教授に対し深く感謝致します。

そして、同研究室皆さんには日頃何かとご協力頂きました。ここに厚くご礼申し上げます。

最後に、本研究の遂行にあたり温かいご激励によって筆者を支えて下さった友人の皆様、そして家族に、心より感謝いたします。